University of Mumbai

वेबसाईट – mu.ac.in
ईमेल - आयडी - dr.aams@fort.mu.ac.in
aams3@mu.ac.in

विद्याविषयक प्राधिकरणे
सभा आणि सेवा विभाग(ए.ए.एम.एस)
रूम नं. १२८ एम.जी.रोड, फोर्ट,
मुंबई – ४०० ०३२
टेलिफोन नं. – ०२२ – ६८३२००३३

(नॅक पुनर्मूल्यांकनाद्वारे ३.६५ (सी.जी.पी.ए.) सह अ++ श्रेणी
विद्यापीठ अनुदान आयोगाद्वारे श्रेणी १ विद्यापीठ दर्जा)

क्र.वि.प्रा.स.से./आयसीडी/२०२५-२६/३७ दिनांक : २७ मे, २०२५

परिपत्रक:-

सर्व प्राचार्य/संचालक, संलग्नित महाविद्यालये/संस्था, विद्यापीठ शैक्षणिक विभागांचे संचालक/ विभाग प्रमुख यांना कळविण्यात येते की, राष्ट्रीय शैक्षणिक धोरण २०२० च्या अमंलबजावणीच्या अनुषंगाने **शैक्षणिक वर्ष २०२५-२६** पासून **पदवी व पदव्युत्तर अभ्यासक्रम** विद्यापरिषदेच्या दिनांक २८ मार्च २०२५ व २० मे, २०२५ च्या बैठकीमध्ये मंजूर झालेले सर्व अभ्यासक्रम मुंबई विद्यापीठाच्या www.mu.ac.in या संकेत स्थळावर NEP २०२० या टॅब वर उपलब्ध करण्यात आलेले आहेत.

मुंबई – ४०० ०३२
२७ मे, २०२५

(डॉ. प्रसाद कारंडे)
कुलसचिव

Scanned with OKEN Scanner

Copy forwarded for information and necessary action to :-	
1	The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Dept)(AEM), dr@eligi.mu.ac.in
2	The Deputy Registrar, Result unit, Vidyanagari drresults@exam.mu.ac.in
3	The Deputy Registrar, Marks and Certificate Unit,. Vidyanagari dr.verification@mu.ac.in
4	The Deputy Registrar, Appointment Unit, Vidyanagari dr.appointment@exam.mu.ac.in
5	The Deputy Registrar, CAP Unit, Vidyanagari cap.exam@mu.ac.in
6	The Deputy Registrar, College Affiliations & Development Department (CAD), deputyregistrar.uni@gmail.com
7	The Deputy Registrar, PRO, Fort, (Publication Section), Pro@mu.ac.in
8	The Deputy Registrar, Executive Authorities Section (EA) eau120@fort.mu.ac.in He is requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to the above circular.
9	The Deputy Registrar, Research Administration & Promotion Cell (RAPC), rapc@mu.ac.in
10	The Deputy Registrar, Academic Appointments & Quality Assurance (AAQA) dy.registrar.tau.fort.mu.ac.in ar.tau@fort.mu.ac.in
11	The Deputy Registrar, College Teachers Approval Unit (CTA), concolsection@gmail.com
12	The Deputy Registrars, Finance & Accounts Section, fort draccounts@fort.mu.ac.in
13	The Deputy Registrar, Election Section, Fort drelection@election.mu.ac.in
14	The Assistant Registrar, Administrative Sub-Campus Thane, thanesubcampus@mu.ac.in
15	The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan, ar.seask@mu.ac.in
16	The Assistant Registrar, Ratnagiri Sub-centre, Ratnagiri, ratnagirisubcentar@gmail.com
17	The Director, Centre for Distance and Online Education (CDOE), Vidyanagari, director@idol.mu.ac.in
18	Director, Innovation, Incubation and Linkages, Dr. Sachin Laddha pinkumanno@gmail.com
19	Director, Department of Lifelong Learning and Extension (DLLE), dlleuniversityofmumbai@gmail.com

Copy for information :-	
1	P.A to Hon'ble Vice-Chancellor, vice-chancellor@mu.ac.in
2	P.A to Pro-Vice-Chancellor pvc@fort.mu.ac.in
3	P.A to Registrar, registrar@fort.mu.ac.in
4	P.A to all Deans of all Faculties
5	P.A to Finance & Account Officers, (F & A.O), camu@accounts.mu.ac.in

To,

1	The Chairman, Board of Deans pvc@fort.mu.ac.in
2	**Faculty of Humanities,** **Offg. Dean** 1. Prof.Anil Singh Dranilsingh129@gmail.com **Offg. Associate Dean** 2. Prof.Manisha Karne mkarne@economics.mu.ac.in 3. Dr.Suchitra Naik Naiksuchitra27@gmail.com
	Faculty of Commerce & Management, **Offg. Dean,** 1 Prin.Ravindra Bambardekar principal@model-college.edu.in **Offg. Associate Dean** 2. Dr.Kavita Laghate kavitalaghate@jbims.mu.ac.in 3. Dr.Ravikant Balkrishna Sangurde Ravikant.s.@somaiya.edu 4. Prin.Kishori Bhagat kishoribhagat@rediffmail.com

	Faculty of Science & Technology **Offg. Dean** 1. Prof. Shivram Garje ssgarje@chem.mu.ac.in **Offg. Associate Dean** 2. Dr. Madhav R. Rajwade Madhavr64@gmail.com 3. Prin. Deven Shah sir.deven@gmail.com
	Faculty of Inter-Disciplinary Studies, **Offg. Dean** 1.Dr. Anil K. Singh aksingh@trcl.org.in **Offg. Associate Dean** 2.Prin.Chadrashekhar Ashok Chakradeo cachakradeo@gmail.com 3. Dr. Kunal Ingle drkunalingle@gmail.com
3	Chairman, Board of Studies,
4	The Director, Board of Examinations and Evaluation, dboee@exam.mu.ac.in
5	The Director, Board of Students Development, dsd@mu.ac.in DSW direcotr@dsw.mu.ac.in
6	The Director, Department of Information & Communication Technology, director.dict@mu.ac.in

AC – 20/5/2025
Item No. – 6.31 (N)

As Per NEP 2020

University of Mumbai

Syllabus for Major Vertical – 1, 4, 5 & 6		
Name of the Programme – B.E. (Electronics and Telecommunication Engineering)		
Faculty of Engineering		
Board of Studies in Electronics and Telecommunication Engineering		
U.G. Second Year Programme	**Exit Degree**	**U.G. Diploma in Electronics and Telecommunication Engineering**
Semester		**III**
From the Academic Year		**2025-26**

University of Mumbai

(As per NEP 2020)

Sr. No.	Heading	Particulars
1	**Title of program** **O:_____**	**B.E. (Electronics and Telecommunication Engineering)**
2	**Exit Degree**	**U.G. Diploma in Electronics and Telecommunication Engineering.**
3	**Scheme of Examination** **R: _____**	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination
4	**Standards of Passing** **R: _____**	**40%**
5	**Credit Structure** **R. TEU-560C** **R. TEU-560D**	**Attached herewith**
6	**Semesters**	**Sem. III**
7	**Program Academic Level**	**5.00**
8	**Pattern**	**Semester**
9	**Status**	**New**
10	**To be implemented from Academic Year**	**2025-26**

Sd/-	**Sd/-**	**Sd/-**
Dr. Faruk Kazi	**Dr. Deven Shah**	**Prof. Shivram S. Garje**
BoS-Chairman- Electronics and Telecommunication Engineering	**Associate Dean**	**Dean**
Faculty of Technology	**Faculty of Science & Technology**	**Faculty of Science & Technology**

Preamble

To meet the challenge of ensuring excellence and NEP 2020 policy in engineering education, the issue of quality needs to be addressed, debated, and taken forward systematically. Accreditation is the principal means of quality assurance in higher education. The major emphasis of the accreditation process is to measure the outcomes of the program that is being accredited. In line with this, the Faculty of Science and Technology (in particular Engineering) of the University of Mumbai has taken the lead in incorporating the philosophy of NEP 2020 education in the process of curriculum development.

The second-year engineering course is a core training program to impart scientific and logical thinking training to learners in general, with a choice of course selection from the program core course, multidisciplinary minor, and vocational skill-enhanced course. Simultaneously, the objectives of NEP 2020 demand nurturing the core program and skills required for the Electronics and Telecommunication Engineering Branch of the learner. Keeping this in view, a pool of courses is offered in Core Courses covering fundamentals required to understand core and modern engineering practices and emerging trends in technology. Considering the shift in pedagogy and the convenience of a stress-free learning process, a choice-based subject pool is offered in the coursework under the heads of Electronics and Telecommunication Engineering for open electives and multidisciplinary minor courses in the third and fourth semesters. Essentially, to give a glimpse of trends in the industry under vocational and enhanced skill practices, the pool is offered to nurture and develop creative skills in contemporary industrial practices. Criteria met in the structure is the opportunity for learners to choose the course of their interest in all disciplines.

The Program Core Course Covers Electronics and Telecommunication engineering core courses. Also, OE and MDM where a pool of subjects are given for selection. Considering the present scenario, diverse choices need to be made available to fulfill the expectation of a learner to aspire for a career in the field of current trends of Technology and interdisciplinary research. Ability enhancement can be achieved in Undergraduate training by giving an objective viewpoint to the learning process and transitioning a learner from a rote learner to a creative professional. for the purpose Design Thinking is introduced in the First Semester to orient a journey learner to become a skilled professional. Considering the NEP-2020 structure of award of Certificate & Diploma at multiple exit-point pools of Vocational skills is arranged for giving exposure to the current Industry practices.

The faculty resolved that course objectives and course outcomes are to be clearly defined for every course so that all faculty members in affiliated higher education institutes understand the depth and approach of the course to be taught, which will enhance the learner's learning process. NEP 2020 grading system enables a much-required shift in focus from teacher-centric to continuous-based learner-centric education since the workload estimated is based on the investment of time in learning and not in teaching. It also focuses on continuous evaluation, which will enhance the quality of education. Credit assignment for courses is based on a 15-week teaching-learning process for NEP 2020, however, the content of courses is to be taught in 12-13 weeks, and the remaining 2-3 weeks are to be utilized for revision, tutorial, guest lectures, coverage of content beyond the syllabus, etc.

There was a concern that in the present system, the second-year syllabus must not be heavily loaded to the learner and it is of utmost importance that the learner entering into the second year of an engineering course should feel at ease by lowering the burden of syllabus and credits. This is necessary for a learner to get accustomed to the new environment of a college and to create a bond between the teacher and the learner. The present curriculum will be implemented for the Second Year of Engineering from the academic year 2025-26. Subsequently, this system will be carried forward for Third Year and Final Year Engineering in the academic years 2026-27, and 2027-28, respectively.

Sd/-	**Sd/-**	**Sd/-**
Dr. Faruk Kazi	**Dr. Deven Shah**	**Prof. Shivram S. Garje**
BoS-Chairman-	**Associate Dean**	**Dean**
Electronics and Telecommunication Engineering	**Faculty of Science & Technology**	**Faculty of Science & Technology**
Faculty of Technology		

Under Graduate Diploma in Engineering- Electronics and Telecommunication.

Credit Structure (Sem. III)

	R. TEU-560C								
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC,RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
	III	PCC301:3 PCC302:3 PCC303:3 PCC304:3 PCL301: 1 PCL302:1	--	--	OE:2	--	VEC: 2 HSL: 2	CEP: 2	22	
	R. TEU-560D									
5.0	IV	PCC401:3 PCC402:3 PCC403:3 PCL401:1 PCL402:1	--	MDM: 4	OE:2	VSEC:2	VEC: 2 EEM:2	--	23	**UG Diploma 45**
	Cum Cr.	25	--	4	4	2	*2+2+2+2*	2	45	

Exit option: Award of UG Diploma in Major and MDM with 90 credits and additional 4 credits core **one** theory subject with 3 credits and **one** lab with 1 credit from one third year from where they want to take Exit degree. Along with theory and practical course student must compulsory do internship for **one month or 160 hours** which internship is equal to 4 credits.

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

S.E. Electronics and Telecommunication Engineering Scheme

Semesters III and IV

Program Structure for Second Year of Electronics and Telecommunication Engineering

UNIVERSITY OF MUMBAI (With Effect from 2025-2026)

SEMESTER III

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2303111	Mathematics for Signal Analysis	2	--	1-	2	1	–	3
2303112	Electronic Devices & Linear Circuits	3	–	--	3	–	–	3
2303113	Digital System Design	3	--	--	3	–	–	3
2303114	Network Theory and Control System	3	--	--	3	–	–	3
OEC301	Open Elective	2#	--	--	2	–	–	2
2303115	Electronic Devices & Linear Circuits Laboratory	--	2	–	–	–	1	1
2303116	Digital System Design Laboratory	--	2	–	–	–	1	1
2303611	C++ and Java Programming	--	2*+2	–	–	–	2	2
2993511	Entrepreneurship Development	--	2*+2	---	–	–	2	2
2993512	Environmental Science	--	2*+2	--	–	–	2	2
Total		**13**	**16**	**01**	**13**	**01**	**08**	**22**

* Two hours of practical class to be conducted for full class as demo/discussion.

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

Institute shall offer a course for Open Elective from Science/Commerce/Management stream bucket provided by the University of Mumbai.

Course Code	Course Description	Examination scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2303111	Mathematics for Signal Analysis	20	20	40	60	2	25	--	125
2303112	Electronic Devices & Linear Circuits	20	20	40	60	2	--	--	100
2303113	Digital System Design	20	20	40	60	2	--	--	100
2303114	Network Theory and Control System	20	20	40	60	2	--	--	100
OEC301	Open Elective	20	20	40	60	2	--	--	100
2303115	Electronic Devices & Linear Circuits Laboratory	--	--	--	--	--	25	25	50
2303116	Digital System Design Laboratory	--	--	--	--	--	25	25	50
2303611	C++ and Java Programming	--	--	--	--	--	50	25	75
2993511	Entrepreneurship Development	--	--	--	--	--	50	--	50
2993512	Environmental Science	--	--	--	--	--	50	--	50
Total		**100**	**100**	**200**	**300**	**10**	**225**	**75**	**800**

Vertical – 1
Major

Detail Syllabus

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2303111	Mathematics for Signal Analysis	2	-	1	2	-	1	3

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Total					
2303111	Mathematics for Signal Analysis	20	20	40	60	2	25	--	125

Rationale:
The goal of this course is to make the learner conversant with the basic tools of mathematics for application in Electrical, Electronics, and Telecommunication engineering. The syllabus designed will help the learner build a foundation to model Signal Analysis problems mathematically, analyze and solve the same.

Prerequisite:
Applied Mathematics-I
Applied Mathematics-II

Course Objectives:

1. To introduce the concept of Laplace Transform and its application in solving ODE.
2. To familiarize with the concept of expanding periodic functions/signals in the form of Fourier Series.
3. To introduce the concept Fourier Transform and its applications.
4. To familiarize with the concept of Z-Transform for discrete functions/signals and its applications.
5. To familiarize with the concept of random variable and probability distributions with its applications in engineering and science.
6. To introduce concepts and fundamentals of Matrix algebra for engineering problems.

Course Outcomes:
On successful completion of the course learner will be able to:

1. Understand Laplace Transform and its application in solving ordinary differential equations.
2. Apply the Fourier series to expand the given periodic function/signal.
3. Apply Fourier Transform and its properties to transform the function/signal from one domain (time) to another domain(frequency).
4. Understand and apply Z-transform to discrete functions/signals.

5. Understand and apply the concept of random variable and standard probability distributions.

6. Apply the concepts of eigenvalues and eigenvectors in engineering problems.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Laplace Transform**	Laplace Transform & Inverse Laplace Transforms of Standard Functions like e^{at}, $\sin(at)$, $\cos(at)$, $\sinh(at)$, $\cosh(at)$and t^n, $where\ n \geq 0$ (without proof) First Shifting theorem, Laplace Transform of derivatives and integrals (Properties without proof) Inverse Laplace transform using First Shifting Theorem and Partial fractions method. Applications of Laplace Transforms for Solutions to ODE to electrical & electronics circuit problems. (Only first order differential equations). **Self -Learning Topics:** Heaviside &Dirac Delta function, Applications of Laplace Transforms for Solutions to ODE (Higher order differential equations)	**5**	**CO1**
II	**Fourier Series**	Fourier series of periodic function with period 2π and 2l. Fourier series of even and odd functions (-l, l), Complex form of Fourier Series$(-\infty, \infty)$(No deductions on the basis of Fourier Series) **Self-learning Topics:** Parseval's Identity, Half-range Cosine/sine Series, Fourier Integral	**4**	**CO2**
III	**Fourier Transform**	Fourier transform, Fourier Transform of Heaviside Unit step Function and Dirac Delta Function. Linearity Property, Time shifting Property, Frequency Shifting Property, convolution and Modulation property, Question related to (Electrical & Extc engineering) **Self -Learning Topics:** Standard Signals-Stop, input, Delta, Exponential Signals	**5**	**CO3**
IV	**Z-Transform**	Definition and Region of Convergence, Transform of Standard Functions: $\{k^n a^k\}$, $\{a^k\}$, $\{c^k \sin(\alpha k + \beta)\}$, $\{c^k sinh\alpha k\}$, $\{c^k \cos(\alpha k + \beta)\}$, $\{c^k cosh\alpha k\}$. Properties of Z Transform: Change of Scale, Shifting Property, Multiplication, and Division by k, Convolution theorem. Inverse Z transform: Partial Fraction Method, Question related to (Electrical & Extc engineering) **Self-learning Topics:** Initial value theorem, Final value theorem, Inverse Using Convolution Theorem	**4**	**CO4**
V	**Random Variable & Probability Distribution**	Discrete and Continuous random variables, Probability mass and density function, Probability distribution for random variables, Expectation, Variance, Poisson and Normal distribution (No question on finding the mean and variance)	**4**	**CO5**

		Self-Learning Topics: Binomial Distribution, Moment generating function		
VI	**Linear Algebra (Theory of Matrices)**	Characteristic Equation, Eigenvalues and Eigenvectors, and properties of eigenvalues (without proof) Similarity of matrices, diagonalizable and non-diagonalizable matrices **Self-learning Topics**: Cayley-Hamilton Theorem and its usage in reduction of higher degree polynomials Derogatory and non-derogatory matrices, Function of a square Matrix	**4**	**CO6**

Note:

- **Tutorial shall be conducted batch wise.**
- **No Questions to be asked from Self-Learning Topics.**

Text / Reference Books:

1. Higher Engineering Mathematics, Dr. B. S. Grewal, Khanna Publication.
2. Advanced Engineering Mathematics, Erwin Kreyszig, Wiley Eastern Limited.
3. Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Narosa publication
4. Signals and Systems, A Nagoor Nani, Tata McGraw Hill.
5. Probability, Statistics and Random Processes, T. Veerarajan, Mc. Graw Hill education.

Online References:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/111/106/111106139/
2.	https://www.youtube.com/watch?v=2CP3m3EgL1Q
3.	https://www.youtube.com/watch?v=Hw8KHNgRaOE

Term Work:

General Instructions:

1. Batch wise tutorials are to be conducted. The number of students per batch should be as per the university pattern for practical.

2. Students must be encouraged to write at least 6 class tutorials on entire syllabus.

3. A group of 4-6 students should be assigned a self-learning topic. Students should prepare a presentation/problem solving of 10-15 minutes. This should be considered as mini project in Engineering mathematics. This project should be graded for 10 marks depending on the performance of the students.

The distribution of Term Work marks will be as follows –

1.	Attendance (Theory and Tutorial)	05 marks
2.	Class Tutorials on entire syllabus	10 marks
3.	Mini project	10 marks

Assessment:

Internal Assessment (IA) Test:

Assessment consists of two class tests of 20 marks each. The first class test (Internal Assessment I) is to be conducted when approx. 40% syllabus is completed and second class test (Internal Assessment II) when additional 35% syllabus is completed. Duration of each test shall be one hour.

End Semester Examination:

1. Question paper will comprise of total 06 questions, each carrying 15 marks.

2. Total 04 questions need to be solved.

3. Question No: 01 will be compulsory and should cover maximum content of the entire syllabus.

4. Remaining questions will be randomly selected from all the modules as per the weightage of each module (which is proportional to number of respective lecture hours mentioned in the syllabus).

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2303112	Electronic Devices and Linear Circuits	3	-	-	3	-	-	3

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Total					
2303112	Electronic Devices and Linear Circuits	20	20	40	60	2	--	--	100

Prerequisite:

1. Engineering Physics-I
2. Engineering Physics-II
3. Basic Electrical Engineering

Course Objectives:

1. To explain functionality of different electronic devices.
2. To perform DC and AC analysis of small signal amplifier circuits.
3. To explain working of differential amplifiers and its applications in operational amplifiers.
4. To understand the concept working principles of Linear Integrated Circuits.
5. To Perform analysis of Linear Integrated Circuits.
6. To design circuits and systems for particular applications using Linear Integrated Circuits.

Course Outcomes:

After successful completion of the course student will be able to

1. Explain working of various electronics devices.
2. Derive expressions for performance parameters of BJT and MOSFET circuits.
3. Understand the fundamentals and areas of applications for the Integrated circuits.
4. Develop the ability to design Linear and Non-Linear application of Integrated Circuits.
5. Cultivate the skill of designing Timer circuits.
6. Gain the skill to design Voltage regulator using Integrated Circuits.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Introduction of Diode, BJT, JFET, MOSFET	01	CO1
I	**Biasing of BJT and MOSFET**	1.1 Construction, working and characteristics of BJT (CE configuration) and E-MOSFET (CS configuration). 1.2 Concept of DC load line, Q point and regions of operations, Biasing circuits for BJT (Fixed bias & Voltage divider Bias). 1.3 DC load line and region of operation for E-MOSFET, Biasing circuits for E-MOSFET (Drain to Gate bias & voltage divider bias).	06	CO1
II	**Small Signal Analysis of BJT and MOSFET Amplifier.**	2.1 Concept of AC load line and Amplification, Small signal analysis (Zi, Zo, Av and Ai) of CE amplifier using hybrid pi model. 2.2 Small signal analysis (Zi, Zo, Av) of CS (for E-MOSFET) amplifiers. 2.3 Frequency response of amplifier, Effect of coupling bypass and parasitic capacitor on frequency response. Millers theorem.	06	CO2
III	**Introduction to Differential Amplifier and Operational Amplifier.**	3.1 E-MOSFET Differential Amplifier, Differential and common mode gain, CMRR, differential and common mode input impedance. 3.2 Block diagram of Op-Amp , Ideal and Practical characteristics of Op-Amp. Open loop and Closed loop configuration of Op-Amp.. 3.3 Inverting and Non-inverting Amplifier using Op-Amp, Summing Amplifier, Difference Amplifier.	06	CO3
IV	**Linear and Non-Linear Applications of Operational Amplifier**	4.1 Integrator & differentiator (ideal & practical), Active Filters: First and Second order active low pass, high pass. 4.2 Comparators: Inverting comparator, non-inverting comparator. Schmitt Triggers: Inverting Schmitt trigger, non-inverting Schmitt trigger. 4.3 Positive feedback, Barkhausen's criteria, Sine Wave Oscillators: RC phase shift oscillator, Wien bridge oscillator.	08	CO4

V	**Timer IC555 and its applications.**	5.1 IC 555 Timer: Block Schematic, Functional Diagram, Working of IC 555. 5.2 Design of Monostable and Astable multivibrator using IC 555. 5.3 Applications of astable and monostable multivibrator as Pulse Width Modulator and Pulse Position Modulator.	06	CO5
VI	**Voltage Regulators.**	6.1 Block diagram of regulated DC power supply. Functional block diagram, working and design of three terminal fixed voltage regulators (78XX, 79XX series). 6.2 Functional block diagram, working and design of general purpose IC 723 (HVLC and HVHC). 6.3 Design of regulator using three terminal IC LM 317.	06	CO6

Text Books:

1. Donald A. Neamen, "Electronic Circuit Analysis and Design", Tata McGraw Hill, 2nd Edition

2. D. Roy Choudhury and S. B. Jain, "Linear Integrated Circuits", New Age International Publishers, 4th Edition.

3. Ramakant A. Gaikwad, " Op-Amps and Linear Integrated Circuits", Pearson Prentice Hall, 4th Edition

References:

1. S. Salivahanan, N. Suresh Kumar, "Electronic Devices and Circuits", Tata Mc-Graw Hill, 3rd Edition
2. Boyiestad and Nashelesky, " Electronic Devices and Circuits Theory", Pearson Education,11th Edition
3. A.K. Maini, " Electronic Devices and Circuits", Wiley
4. K.R. Botkar, "Integrated Circuits", Khanna Publisher (2004)
5. David A. Bell, " Operation Amplifiers and Linear Integrated Circuits", Oxford University Press, Indian Edition.

Online References:

Sr. No.	**Website Name**
1.	NPTEL/ Swayam Course: Course: Integrated Circuits and Applications By Prof. Shaik Rafi Ahamed (IIT Guwahati) https://onlinecourses.nptel.ac.in/noc25_ee43/preview

2.	Course: ICs MOSFETs Op-Amps & Their Applications By Prof. Hardik Jeetendra Pandya (IISc Bangalore); https://swayam.gov.in/nd1_noc20_ee13/preview

Assessment:

Internal Assessment (IA) Test:

Assessment consists of two class tests of 20 marks each. The first class test (Internal Assessment I) is to be conducted when approx. 40% syllabus is completed and second class test (Internal Assessment II) when additional 35% syllabus is completed. Duration of each test shall be one hour.

End Semester Examination:

1. Question paper will comprise of total 06 questions, each carrying 15 marks.
2. Total 04 questions need to be solved.
3. Question No: 01 will be compulsory and should cover maximum content of the entire syllabus.
4. Remaining questions will be randomly selected from all the modules as per the weightage of each module (which is proportional to number of respective lecture hours mentioned in the syllabus).

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2303113	Digital System Design	3	--	--	3	--	--	3

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Total					
2303113	Digital System Design	20	20	40	60	2	--	--	100

Prerequisite:
Basic Electrical Engineering

Course Objectives:

1. To understand number system representations and their inter-conversions used in digital electronic circuits.
2. To understand the functionalities, and Characteristics of Logic Families and Minimization techniques to realise logical operations.
3. To analyze digital logic processes and to implement logical operations using various combinational logic circuits.
4. To analyze, design and implement the logical operations using different sequential logic circuits.
5. To equip students with the knowledge and skills to design, and implement various registers, counters, and programmable logic devices.
6. To get acquainted with the basics of VHDL language.

Course Outcomes:

On successful completion of the course student will be able to:

1. Apply the concepts of number systems and perform code conversions.
2. Classify logic families, Understand Digital circuits and apply minimization techniques to implement logical functions.
3. Analyze, design and implement combinational logic circuits.
4. Analyze, design and implement sequential logic circuits.
5. Analyze, design and implement digital circuits using different registers, counters, and programmable logic devices.
6. Use HDL & appropriate EDA tool for logic design and simulation using VHDL/Verilog.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Number Systems and Codes**	Review of Binary, Octal and Hexadecimal Number Systems, their inter-conversion, Gray code and BCD code, Binary Addition, Subtraction using 1's and 2's Complement method.	**02**	CO1
II	**Logic families and Minimization Techniques**	Classification of logic families: Unipolar and Bipolar Logic Families, Characteristics of Digital ICs, TTL and CMOS comparison. Digital logic gates, Universal gates, Realization using NAND and NOR gates, Boolean Algebra, De Morgan's Theorem. Minimization of Boolean expressions :- SOP, POS, and Karnaugh map (up to 4 variables)	**08**	CO2
III	**Combinational Logic Circuits**	Adder, Subtractor, Multiplexer, De-multiplexer, Code Converter, BCD adder, Magnitude Comparator, Parallel Adder, Implementation of Logic expressions using Multiplexers, De-multiplexers, Encoders and Decoders.	**08**	CO3
IV	**Sequential Logic Circuits**	Flip flops (FF): SR, JK, T, D, Master Slave JK flip flops, Truth table, excitation table, triggering methods, and flip flop conversions. Counters: Asynchronous and Synchronous - MOD N, UP/DOWN, Decade counter, Frequency division, Finite State Machine: Introduction to Moore and Mealy machines - Block diagram, state diagram, state tables.	**10**	CO4
V	**Shift Registers and Programmable Logic Devices**	Registers: SISO, SIPO, PISO, PIPO, Universal Shift registers, Ring counter, Johnson counter, Sequence generator. Structure of Programmable Logic Devices (PLDs), Function implementation with Programmable Logic Array (PLA) and Programmable Array Logic (PAL). Introduction to CPLD and FPGA.	**06**	
VI	**Introduction to VHDL**	VLSI Design flow (Frontend): Design entry: Schematic different modeling styles in VHDL, Data types and objects, Synthesis and Simulation, implementation of combinational and sequential logic using VHDL.	**05**	CO6

Text Books:

1. R.P. Jain, "Modern Digital Electronics", Tata McGraw Hill Publication, 4th Edition.
2. Morris Mano, Michael D. Ciletti, "Digital Design", Pearson Education, Fifth Edition (2013).
3. A. Anand Kumar, "Fundamentals of Digital Circuits", PHI, Fourth Edition (2016).

4. J. Bhaskar A Verilog HDL Primer , Third Edition, Star Galaxy publishing
5. Sameer Palnitkar "Verilog HDL, A guide to digital
6. Douglas Perry, "VHDL programming", McGraw Hill, fourth edition.

References:

1. John F. Warkerly, "Digital Design Principles and Practices", Pearson Education,Fifth Edition (2018).
2. Digital fundamentals by FLOYD & JAIN, Pearsons Pub
3. Charles Roth, "Digital System Design using VHDL", Tata McGraw Hill

Assessment:

Internal Assessment (IA) Test:

Assessment consists of two class tests of 20 marks each. The first class test (Internal Assessment I) is to be conducted when approx. 40% syllabus is completed and second class test (Internal Assessment II) when additional 35% syllabus is completed. Duration of each test shall be one hour.

End Semester Examination:

1. Question paper will comprise of total 06 questions, each carrying 15 marks.

2. Total 04 questions need to be solved.

3. Question No: 01 will be compulsory and should cover maximum content of the entire syllabus.

4. Remaining questions will be randomly selected from all the modules as per the weightage of each module (which is proportional to number of respective lecture hours mentioned in the syllabus).

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2303114	Network Theory and Control System	3	-	-	3	-	-	3

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Total					
2303114	Network Theory and Control System	20	20	40	60	2	--	--	100

Prerequisite:

1. Basic Electrical Engineering
2. Engineering Mathematics II

Course Objectives:

1. To evaluate the Circuits using network theorems, study network Topology, network Functions and two port networks.
2. To analyze the Circuits in time and frequency domain.
3. To synthesize passive network by various methods.
4. To analyze fundamental concepts of mathematical modeling, time response and Frequency response.
5. To develop concepts of stability and its assessment criteria.

Course Outcomes:

After successful completion of the course student will be able to

1. Evaluate circuit using network theorems.
2. Apply the time and frequency method of analysis.
3. Analyze the network function and finding the various parameters of two port network
4. Analyze the response and determine the transfer function of Control System
5. Understand the analysis of systems in time domain and predict stability of given system using appropriate criteria.
6. Understand the analysis of systems in frequency domain and predict stability of given system using appropriate criteria.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Electrical Circuit Analysis**	**I.1)** Analysis of Circuits with dependent sources using generalized loop and node analysis, super mesh and super node analysis technique Network Theorems with dependent sources: Superposition, Thevenin's, Norton's and Maximum Power Transfer Theorems **(Use only DC source)**	4	CO1
II	**Time and frequency domain analysis**	**II.1)** Time domain analysis of R-L and R-C Circuits: Forced and natural response, initial and final values. Solution using first order and second order differential equation with step signals	8	CO 2
		II.2) Frequency domain analysis of R-L-C Circuits: Forced and natural response, effect of damping factor. Solution using second order equation for step signal.		
III	**Network functions and Two Port Networks**	**III.1)** Network functions for the one port and two port networks, driving point and transfer functions, Poles and Zeros of Network functions.	6	CO3
		III.2) Two Port Parameters: Open Circuits, short Circuit, Transmission and Hybrid parameters, relationship among parameters, conditions for reciprocity and symmetry		
IV	**Analysis and response of control system**	**IV.1)** Open and closed loop systems, Transfer function modeling (Electrical only), Block diagram reduction techniques and Signal flow graph.	8	CO4
		IV.2) Dynamic Response: Standard test signals, transient and steady state behavior of first and second order systems, steady state errors in feedback control systems and their types.		
V	**Stability Analysis in Time Domain**	**V.1)** Concept of stability: Routh and Hurwitz stability criterion	6	CO5
		V.2) Root locus Analysis: Root locus concept, general rules for constructing root-locus, root locus analysis of control system.		
VI	**Stability Analysis in Frequency Domain**	**VI.1)** Frequency domain specification, Relationship between time and frequency domain specification of system, stability margins	7	CO6
		VI.2) Bode Plot: Magnitude and phase plot, Method of plotting Bode plot, Stability margins and analysis using Bode plot. Concept of Polar plot		

Textbooks:

1. Franklin F Kuo, “Network Analysis and Synthesis”, Wiley Toppan, 2 nd ed. ,1966.
2. M E Van Valkenburg, “Network Analysis”, Prentice-Hall of India Pvt Ltd, New Delhi, 26th Indian Reprint, 2000.

References:

1. A. Chakrabarti, “Circuit Theory”, Dhanpat Rai & Co., Delhi, 6th Edition.
2. A. Sudhakar, Shyammohan S. Palli “Circuits and Networks”, Tata McGraw-Hill education.
3. Smarajit Ghosh “Network Theory Analysis & Synthesis”, PHI learning.
4. K.S. Suresh Kumar, “Electric Circuit Analysis” Pearson, 2013.
5. D. Roy Choudhury, “Networks and Systems” , New Age International, 1998.
6. Nagrath, M.Gopal, “Control System Engineering”, Tata McGrawHill.
7. Rangan C. S., Sarma G. R. and Mani V. S. V., "Instrumentation Devices And Systems", Tata McGraw-Hill, 2nd Ed.,2004.
8. K.Ogata, “Modern Control Engineering, Pearson Education”, IIIrd edition.

NPTEL / Swayam Course:

1. Course: Basic Electrical Circuits By Prof. Nagendra Krishnapura (IIT Madras); https://swayam.gov.in/nd1_noc20_ee64/preview.
2. Course: Control Systems By Prof. C. S. Shankar Ram (IIT Madras); https://swayam.gov.in/nd1_noc20_ee90/preview

Assessment:

Internal Assessment (IA) Test:

Assessment consists of two class tests of 20 marks each. The first class test (Internal Assessment I) is to be conducted when approx. 40% syllabus is completed and second class test (Internal Assessment II) when additional 35% syllabus is completed. Duration of each test shall be one hour.

End Semester Examination:

1. Question paper will comprise of total 06 questions, each carrying 15 marks.

2. Total 04 questions need to be solved.

3. Question No: 01 will be compulsory and should cover maximum content of the entire syllabus.

4. Remaining questions will be randomly selected from all the modules as per the weightage of each module (which is proportional to number of respective lecture hours mentioned in the syllabus).

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2303115	Electronic Devices & Linear Circuits Laboratory	--	2	--	--	1	--	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg. of 2 Tests				
2303115	Electronic Devices & Linear Circuits Laboratory	--	--	--	--	25	25	50

Laboratory Objectives:

1. To make students familiar with equipment and measuring instruments used to perform this laboratory course.
2. To provide hands on experience to develop laboratory setup for performing given experimental using various equipment, electronic devices and measuring instruments.
3. To develop an ability among students to gather appropriate data and analyse the same to relate theory with practical.
4. To develop trouble shooting abilities among students

Laboratory Outcomes:

After successful completion of the course student will be able to

1. Know various equipment used in this laboratory course.
2. Understand how to make use of various devices and equipment to perform laboratory work.
3. Perform given experiment by making proper connections between various components, equipment and measuring devices for this course.
4. Acquire requisite data and analyze the same for this course.
5. Evaluate various parameters of the given circuit for this course.
6. Design the circuit for a given application for this course.

Suggested List of Experiments:

Sr No	List of Experiments	Hrs.
01	To study BJT biasing Circuits.	2
02	To Study BJT as CE amplifier.	2
03	To study EMOSFET biasing circuits	2
04	To study EMOSFET as CS amplifier.	2

05	**Simulations Experiment on study of Frequency Response of CS amplifier.**	**2**
06	**Simulations Experiment on study of Differential amplifier**	**2**
07	**Design and Implementation of Adder circuits using OPAMP.**	**2**
08	**Design and Implementation of Difference Amplifier using OPAMP.**	**2**
09	**Design and analyze Integrator circuit using OPAMP.**	**2**
10	**Design and analyze Differentiator circuit using OPAMP.**	**2**
11	**Design and analyze Schmitt trigger using OPAMP.**	**2**
12	**Design and analyze RC phase shift Oscillator.**	**2**
13	**Design and analyze first order High pass and Low pass filter.**	**2**
14	**Design of Monostable Multivibrator Circuit using 555 Timer**	**2**
15	**Design of Astable Multivibrator Circuit using 555 Timer**	**2**

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2303116	Digital System Design Laboratory	--	2	--	--	1	--	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg. of 2 Tests				
2303116	Digital System Design Laboratory	--	--	--	--	25	25	50

Laboratory Objectives:

1. To get familiarise with basic building blocks of Digital System Design and verify the operation of various digital ICs.
2. To understand and implement digital circuits for code conversion.
3. To train students to design and implement combinational circuits.
4. To instruct students on how to design and implement sequential circuits.
5. To understand digital logic simulation using the EDA tool.

Laboratory Outcomes:

After successful completion of the course student will be able to

1. Identify various Digital ICs and basic building blocks of digital system design
2. Design and implement combinational circuits like adder, subtractor, multiplexer, code converters etc.
3. Identify and understand the working of various types of flip flops and their interconversions.
4. Design and implement basic sequential circuits such as counters, registers etc
5. Develop and simulate VHDL architectural representations of digital systems and components using structural, behavioural, or data flow concepts

Suggested List of Experiments:

Sr No	List of Experiments	Hrs.
01	Study of characteristics of typical TTL and CMOS IC's like fan out, noise margin, propagation delay.	02
02	Implement AND, OR, NOT, EXOR, EX-NOR gates using Universal gates NAND and NOR.	02
03	Simplify the logical expressions using Boolean algebra/k-map technique and implement using logic gates.	02

04	Implement digital circuits to perform code conversions like Binary to Gray and Gray to Binary, BCD to 7 segment decoder operations.	02
05	Design and implement Encoder/ Decoder using IC.	02
06	Design and implement logic equations using Multiplexer IC.	02
07	Flip-flop conversions JK to D, JK to T and D to T FF.	02
08	Design and implementation of ripple and synchronous counters using JK and D FF and additional gates.	02
09	Design of counter using ICs like 7490/93 (ripple) and 74192/193(synchronous)	02
10	Study of Universal Shift Register using IC-74194.	02
11	Design a Ring/ Johnson's counter using IC-74194.	02
12	Implement a universal gates using VHDL/Verilog	02
13	Implement adder circuits using VHDL/Verilog	02
14	Design a Multiplexer using VHDL/Verilog	02
15	Design a 3-bit linear feedback shift register (LFSR) using VHDL/Verilog	02
16	Design a 3-bit Array Multiplier using VHDL/Verilog	02
17	Design a 2-bit Vedic Multiplier using VHDL/Verilog	02
18	Design and implementations of random sequence counter using D FF or JK FF ICs	02
19	Comparator using IC 7485 and Parity generator and checker using X-OR gate	02
20	Binary and BCD adders and Subtractor using IC 7483 and gates	02
21	Design asynchronous/synchronous MOD N counter using IC7490	02
22	Design and implement Magnitude Comparator.	02

Sr. No.	List of Assignments / Tutorials	Hrs.
01	Number Systems and Interconversions, Binary Codes.	01
02	Boolean Algebra and Minimization using K-Map.	01
03	Digital logic gates, Universal gates, Realization using NAND and NOR gates.	01
04	Design of Combinational and Sequential Logic Circuits.	01
05	PLDs and VHDL.	01

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Vertical – 4

Vocational and Skill Enhancement Course (VSEC)

Detail Syllabus

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2303611	C++ and Java Programming	-	4	-	-	2	-	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg. of 2 Tests				
2303611	C++ and Java Programming	--	--	--	--	50	25	75

Prerequisite:
C-Programming

Laboratory Objectives:

1. To introduce Object-Oriented Programming (OOP) principles and understand the necessity of OOP in software development using C++ and Java.
2. To develop problem-solving skills using control structures, functions, arrays, strings, and object-oriented programming concepts in C++.
3. To implement concepts like inheritance, polymorphism, operator overloading, file handling, and memory management in C++ and Java for better software design.
4. To explore Java programming paradigms and understand its differences from C++, focusing on Java classes, methods, inheritance, and polymorphism.
5. To familiarize students with advanced Java concepts like exception handling, multithreading, GUI programming, and applet development.

Laboratory Outcomes:

After successful completion of the course student will be able to

1. Demonstrate basic programming constructs such as data types, control statements, arrays, and strings in C++ and Java.
2. Apply object-oriented programming concepts such as classes, objects, encapsulation, inheritance, and polymorphism in software design.
3. Implement operator overloading, file handling, constructors, and destructors in C++ for efficient memory and resource management.
4. Develop Java applications using classes, objects, interfaces, exception handling, multithreading, and GUI programming.
5. Design and implement applet-based applications and GUI-based Java programs using AWT and event handling techniques.

DETAILED SYLLABUS

Sr. No.	Module	Detailed Content	Hours	LO Mapping
I		**Overview of CPP**	4	LO1
	1.1.	Need of Object-Oriented Programming (OOP), Object Oriented Programming Paradigm, Basic Concepts of Object-Oriented Programming, Benefits of OOP and C++ as object-oriented programming language.		
	1.2	C++ programming Basics, Data Types, Structures, Enumerations, control structures, Class, Object, class and data abstraction, class scope and accessing class members, separating interface from implementation, controlling access to members.		
		Prerequisites: **Students should have knowledge of** 1. **Basic Computer Knowledge** o Understanding how software works, how to run programs, compile code. 2. **Introduction to Programming** o Basic syntax and structure of C programming language o Familiarity with writing simple programs (input/output, variables, loops). 3. **Fundamentals of C Programming (Optional but helpful)** o Data types, variables, control structures (if, for, while). o Functions and arrays. **Self-Learning Topics** **Branching -** If statement, If-else Statement, Decision. **Looping –** while, do-while, for loop **Nested control structure-** Switch statement, Continue statement, Break statement.		
II		**C++ Function, Array and Strings**	6	LO1
	2.1	Returning values from functions. Reference arguments. Overloaded function. Inline		

		function. Default arguments. Return by reference		
	2.2	**Array and Strings** Concepts, Declaration, Definition, Accessing array element, One-dimensional and Multidimensional array. String, String Functions, standard C++ String class		
III		**Object-Oriented Programming using C++ and Files**	8	LO2, LO3
	3.1	**Operator Overloading-** concept of overloading, operator overloading, Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Type casting (implicit and explicit), Pitfalls of Operator Overloading and Conversion, Keywords explicit and mutable. **Function-** Function prototype, accessing function and utility function, Constructors and destructors, Copy Constructor, Objects and Memory requirements, Static Class members, data abstraction and information hiding, inline function. **Constructor-** Definition, Types of Constructors, Constructor Overloading, Destructor.		
	3.2	**Inheritance-** Introduction, Types of Inheritance, Inheritance, Public and Private Inheritance, Multiple Inheritance, Ambiguity in Multiple Inheritance, Visibility Modes Public, Private, Protected and Friend, Aggregation, Classes Within Classes. Deriving a class from Base Class, Constructor and destructor in Derived Class, Overriding Member Functions, Class Hierarchies, **Polymorphism-** concept, relationship among objects in inheritance hierarchy, Runtime & Compile Time Polymorphism, abstract classes, Virtual Base Class.		
	3.3.	**File -**Stream in CPP, Class for File Stream Operation, Modes of Files, Opening and Closing File, Read, Write and append in File**.**		
IV		**Introduction to Java**	2	LO4
	4.1	Programming paradigms- Introduction to programming paradigms, Introduction to four main Programming paradigms like procedural, object oriented, functional, and logic & rule		

		based. Difference between C++ and Java		
	4.2	Java History, Java Features, Java Virtual Machine, Data Types and Size (Signed vs. Unsigned, User Defined vs. Primitive Data Types, Explicit Pointer type), Programming Language JDK Environment and Tools.		
V		**Inheritance, Polymorphism, Encapsulation using Java**	8	LO4
	5.1	**Classes and Methods**: class fundamentals, declaring objects, assigning object reference variables, adding methods to a class, returning a value, constructors, this keyword, garbage collection, finalize () method, overloading methods, argument passing, object as parameter, returning objects, access control, static, final, nested and inner classes, command line arguments, variable-length Arguments.		
	5.2.	Array, String, String buffer and Vectors		
	5.3	**Inheritances:** Member access and inheritance, super class references, Using super, multilevel hierarchy, constructor call sequence, method overriding, dynamic method dispatch, abstract classes, Object class. **Packages and Interfaces:** defining a package, finding packages and CLASSPATH, access protection, importing packages, interfaces (defining, implementation, nesting, applying), variables in interfaces, extending interfaces, instance of operator.		
VI		**Exception Handling and Applets in Java**	8	LO4, LO5
	6.1	**Exception Handling:** fundamental, exception types, uncaught exceptions, try, catch, throw, throws, finally, multiple catch clauses, nested try statements, built-in exceptions, custom exceptions (creating your own exception sub classes). **Multithreading Threading:** Introduction, thread life cycle, Thread States: new, runnable, Running, Blocked and terminated, Thread naming, thread join method, Daemon thread		
	6.2.	**Applet:** Applet Fundamental, Applet Architecture, Applet Life Cycle, Applet Skeleton, Requesting Repainting, status window, HTML Applet tag, passing parameters to Applets, Applet and Application Program		

	6.3	**GUI :** Introduction to AWT programming Layout and Component Managers Event handling.		
VII		**Mini Project**	3	LO2, LO3, LO4, LO5

Textbooks:

1. **E. Balagurusamy**, *"Object Oriented Programming with C++"*, Tata McGraw-Hill Education, **Sixth Edition**, 2013.
2. **D. Ravichandran**, *"Programming with C++"*, Tata McGraw-Hill Publishing Company Limited, **Second Edition**, 2006.
3. **Yashavant Kanetkar**, *"Let Us C++"*, BPB Publications, **Revised Edition**, 2020.
4. **E. Balagurusamy**, *"Programming with Java: A Primer"*, Tata McGraw-Hill Education, **Fifth Edition**, 2014.
5. **Cay S. Horstmann**, *"Core Java Volume I – Fundamentals"*, Pearson Education, **Eleventh Edition**, 2018.
6. **Kathy Sierra, Bert Bates**, *"Head First Java"*, O'Reilly Media, **Second Edition**, 2005.

References:

1. **Herbert Schildt**, *"Java: The Complete Reference"*, Tata McGraw-Hill Publishing Company Limited, **Ninth Edition**, 2014.
2. **Bjarne Stroustrup**, *"The C++ Programming Language"*, Addison-Wesley, **Fourth Edition**, 2013.
3. **Stanley B. Lippman, Josée Lajoie, Barbara E. Moo**, *"C++ Primer"*, Addison-Wesley, **Fifth Edition**, 2012.
4. **Scott Meyers**, *"Effective C++: 55 Specific Ways to Improve Your Programs and Designs"*, Addison-Wesley, **Third Edition**, 2005.
5. **Joshua Bloch**, *"Effective Java"*, Addison-Wesley, **Third Edition**, 2018.
6. **Sachin Malhotra, Saurabh Chaudhary** "Programming in Java", Oxford University Press, 2010.
7. **Grady Booch, James Rumbaugh, Ivar Jacobson,** "The Unified Modeling Languageser Guide", Pearson Education.

Software Tools for C++ and Java Programming

SrNo	Software Tools for C++ and Java Programming
1	**Visual Studio Code (VS Code)** o **Platform:** Windows, macOS, Linux o **Languages:** C++, Java (via extensions) o **Features:** Lightweight, IntelliSense, debugging, Git integration, customizable with extensions.
2	**Eclipse IDE** o **Platform**: Windows, macOS, Linux o **Languages:** Primarily Java, with C/C++ support (via CDT plugin)

	○ **Features:** Project management, code analysis, UI builder for Java apps.
3	**NetBeans IDE** ○ **Platform:** Windows, macOS, Linux ○ **Languages:** Java, C, C++ ○ **Features:** Excellent Java support, GUI builder, simple setup for projects
4	**IntelliJ IDEA** ○ **Platform:** Windows, macOS, Linux ○ **Languages:** Java, Kotlin, Scala, C++ (limited) ○ **Features:** Powerful Java IDE with smart code completion and refactoring tools.
5	**Code::Blocks** ○ **Platform:** Windows, macOS, Linux ○ **Languages:** C, C++ ○ **Features:** Lightweight C++ IDE, great for beginners, plugin support.
6	**Dev C++** ○ **Platform:** Windows ○ **Languages:** C, C++ ○ **Features:** Simple IDE with built-in compiler, perfect for learning C++.
7	**BlueJ** ○ **Platform:** Windows, macOS, Linux ○ **Languages:** Java ○ Features: Educational IDE designed for teaching OOP concepts.
8	**JGrasp** ○ **Platform:** Windows, macOS, Linux ○ **Languages:** Java, C, C++ ○ **Features:** Lightweight IDE with visualization tools for Java structures
9	**Xcode** ○ **Platform:** macOS ○ **Languages:** C, C++, Objective-C, Swift, Java (limited) ○ **Features:** Excellent for Apple ecosystem development; supports C++ well
10	**CLion (by JetBrains)** • **Platform:** Windows, macOS, Linux • **Languages:** C, C++ • **Features:** Smart C++ development with debugging, refactoring, CMake support.

Skill-Enhancement Activities for C++ and Java Programming
1. Use of Tools for programming and project development
Objective

- The objective is to enhance students' programming skills using industry-standard tools like Eclipse, NetBeans, IntelliJ IDEA, Code:Blocks, Dev C++, and BlueJ. It aims to familiarize them with project development, debugging, and external library integration. Training on tools like Maven prepares students for real-world software practices through hands-on, project-based learning.

Purpose

- To enhance programming skills and industry readiness, students should be trained to code using a variety of development environments and tools. **Eclipse**, **NetBeans**, and **IntelliJ IDEA** are widely accepted IDEs in the software industry, especially for Java development. These platforms support integration of **external libraries and frameworks**, project structuring, and version control. Students should also be introduced to **Maven** as a powerful build tool for managing project dependencies and builds efficiently.
- For C++ programming, students should gain hands-on experience using **Code::Blocks**, **Dev C++**, and **BlueJ**, which provide user-friendly interfaces and help build strong foundational knowledge in object-oriented programming and file handling. These tools also help in understanding how code compiles, links, and executes, making them ideal for beginners and intermediate learners. Emphasis should be placed on solving real-world problems, debugging, and project-based learning using these tools.

2. Real-World Mini Problem Statements via Industry Simulation

- **Objective:**
  Bridge the gap between academics and industry by simulating real-world development tasks aligned with **company-level expectations**.
- **Implementation Strategy:**
  - Identify **10–15 mini problem statements** from domains such as inventory management, attendance tracking, student feedback systems.
  - Organize students into **groups of 3–4** and assign one problem per group.
  - Allocate **8–12 hours (across the last two lab sessions)** for brainstorming, coding, testing, and demo.
  - Encourage the use of Eclipse, Maven, Git, and external APIs or libraries relevant to the solution.
- **Examples of Problem Statements:**
  - Build a **Leave Management System** using Java classes and file I/O.
  - Develop a **Library Management GUI** with Swing and JDBC.
  - Implement a **Patient Record Tracker** with JSON serialization and external libraries.
  - Create a **Book Recommendation Console App** using OOP and collections.
- **Expected Outcome:**
  - Students experience the **complete software development cycle**—from understanding a requirement to deploying a working solution.
  - Promotes **teamwork, time management, and tool proficiency**, building job-ready skills for campus placements.

Online Resources:

Sr. No.	Website Name (CPP Programming)
1.	**cplusplus.com -** Comprehensive reference for C++ syntax, standard libraries, and STL. Great for quick lookups http://www.cplusplus.com
2.	**GeeksforGeeks(C++) -** Rich in tutorials, quizzes, practice problems, and interview questions. Ideal for beginners to advanced learners https://www.geeksforgeeks.org/c-plus-plus
3.	**Codecademy – Learn C++ -** Interactive platform with real-time coding in browser. Gamified progress and projects included. https://www.codecademy.com/learn/learn-c-plus-plus
4.	**Coursera – C++ For C Programmers -** University-style course, great for structured learners. Includes peer-reviewed assignments and quizzes https://www.coursera.org/learn/c-plus-plus-a
.5.	**Udemy – Beginning C++ Programming - From Beginner to Beyond -** Covers both fundamentals and advanced concepts, Great for beginners and intermediate learners. https://www.udemy.com/course/beginning-c-plus-plus-programming/
6.	**NPTEL – Programming in C++-** Offered by IIT Kharagpur,Free to access, Comprehensive and academic-focused, Includes assignments, weekly quizzes, and final certification exam. https://nptel.ac.in/courses/106/105/106105151/
	Website Name (Java Programming)
1.	**JavaTpoint -** Easy-to-understand explanations, tons of examples, and hands-on exercises. Covers basics to frameworks. https://www.javatpoint.com/java-tutorial
2.	**Oracle Java Documentation -** The official and most authoritative resource on Java. Best for understanding the language in depth. https://docs.oracle.com/javase/tutorial/
3.	**W3Schools Java Tutorial-** Beginner-friendly and offers a try-it-yourself feature to code online. https://www.w3schools.com/java/
4.	**Coursera – Java Programming and Software Engineering Fundamentals -** Offered by Duke University,Beginner-friendly, includes real-world applications like web scraping and data analysis. https://www.coursera.org/specializations/java-programming
5.	**Udemy – Java Programming Masterclass updated to Java –** Over 80 hours of content, covering Java from basics to advanced, Taught by experienced software engineer https://www.udemy.com/course/java-the-complete-java-developer-course/
6.	**NPTEL – Object-Oriented Programming using Java** https://nptel.ac.in/courses/106/105/106105191/ - Offered by IIT Kharagpur, In-depth academic course with real-life applications and Java-specific concepts,Includes weekly assignments, lectures,

List of Experiments.

Problems Statement can be divided in three parts

1. Some Few can be solved in class during Lecture so involvement of Students will increase
2. Some Few can be given as Assignment so that repeated process will retain the syntax and logic
3. Some can be asked to solve while doing the practical session

Note : Out of the given list topic wise 25% can be solved in class, 50% can be taken in Lab and remaining 25% can be given as assignment

Unit	Topic	LO
1.0	1. Develop a C++ program to demonstrate the concept of class and object by creating a simple "Bank Account" management system. 2. Create a program that uses structures and enumerations to store and display student information. 3. Implement a C++ application that showcases the use of encapsulation by creating a class for employee data management. 4. Write a program to demonstrate polymorphism using function overloading and operator overloading. 5. Design a C++ program to calculate the area of different shapes (Circle, Rectangle, Triangle) using function overloading. 6. Create a C++ program to simulate a simple library management system using classes and objects. 7. Implement a program using default arguments and inline functions to calculate the volume of different geometric shapes. 8. Develop a C++ application to demonstrate the use of reference arguments in a function for swapping two numbers. 9. Write a C++ program that uses the concept of separating interface from implementation by creating a class for basic arithmetic operations. 10. Build a simple program to demonstrate returning values by reference in C++ using a class to manage complex numbers. **Objective :** These problem statements and objectives cover various concepts from OOP, including encapsulation, polymorphism, data abstraction, and other C++ programming concepts.	LO1
2.0	**1.** Write a C++ program to find the largest of three numbers	LO1

	using an if-else statement**.** 2. Develop a program to check whether a given number is prime using a while loop**.** 3. Create a C++ program that simulates a simple menu-driven calculator using a switch statement**.** 4. Implement a program to print the Fibonacci series using a do-while loop. 5. Write a C++ program to display numbers from 1 to 100, but skip multiples of 5 using the continue statement. **6.** Develop a C++ program to input and display elements of a one-dimensional array**.** **7.** Write a C++ program to perform matrix addition using a two-dimensional array**.** 8. Create a program to check if a given string is a palindrome using standard C++ string functions. 9. Implement a C++ program to count the number of vowels and consonants in a given string. 10. Design a simple student management system using structures to store student details and display information. **Objective :** To develop problem-solving skills and logical thinking by applying C++ control statements, arrays, and string manipulation techniques to create efficient and optimized programs.	
3.0	1. Create a C++ program to demonstrate operator overloading for adding two complex numbers using the + operator. 2. Implement a program to overload the ++ operator for incrementing the values of a custom class object. 3. Develop a C++ application that demonstrates explicit type casting using constructors. 4. Write a C++ program that demonstrates the use of mutable keyword for modifying a constant object. 5. Design a class with a copy constructor to create a duplicate of an existing object.. 6. Create a program using single inheritance to derive a class Student from a base class Person. 7. Write a C++ program that demonstrates multiple inheritance by creating a class that inherits from two base classes. 8. Develop a program to illustrate the concept of virtual base class to solve the diamond problem. Implement a C++ program that demonstrates runtime polymorphism using virtual functions.. 11. Create a base class Shape with a virtual function draw()	• Practical 1–4 → LO2 • Practical 5 → LO3 • Practical 6–10 → LO2 • Practical 11–15 → LO3

	and derive classes like Circle and Rectangle to override the function. Write a C++ program to read and write data to a file using file streams.. 12. Implement a simple file-handling application to append data to an existing file. 13. Create a student management system where student details are stored and retrieved from a file.. 14. Develop a C++ program to count the number of words in a given text file. 15. Implement a C++ application to copy content from one file to another. **Objective :** To understand and apply object-oriented programming concepts such as operator overloading, inheritance, polymorphism, and file handling in C++ to develop efficient and maintainable applications.	
4.0	1. Develop a Java program to demonstrate the procedural programming approach using simple arithmetic operations. 2. Implement a class in Java to demonstrate object-oriented programming by creating a Student class with attributes and methods. 3. Create a Java program to illustrate functional programming using lambda expressions and streams. 4. Write a simple rule-based expert system in Java using conditional statements to suggest clothing based on weather input. 5. Compare and contrast C++ and Java by implementing a simple calculator program in both languages. 6. Develop a Java program to demonstrate different data types and their sizes using simple variables. 7. Write a Java program to convert an integer from signed to unsigned using bitwise operations. 8. Create a simple Java application using JDK and explain the development process including compiling and running using javac and java commands. 9. Develop a Java program to illustrate the concept of explicit pointers using references. 10. Implement a Java program that simulates the working of a Java Virtual Machine (JVM) by creating and running multiple threads. **Objective :** To understand the fundamentals of Java programming, including programming paradigms, Java history,	LO4

	features, data types, JVM functionality, and the use of JDK tools for developing efficient applications.	
5.0	1. Write a Java program to demonstrate class fundamentals by creating a class Employee with attributes and methods. 2. Develop a Java program to demonstrate the use of this keyword to differentiate between instance variables and parameters. 3. Create a Java program that uses static methods and variables to calculate the area of a rectangle. 4. Implement a Java program to demonstrate method overloading using different parameter types. 5. Write a program to demonstrate the use of garbage collection and the finalize() method. 6. Develop a Java program to create and manipulate arrays by finding the largest element. 7. Write a Java program to perform basic string operations using the String class. Create a Java program to demonstrate the use of StringBuffer to reverse a given string. 8. Implement a program to simulate a simple task management application using Vectors. 9. Write a Java program to concatenate two strings using StringBuilder.. 11. Develop a Java program to demonstrate single inheritance using a Person class and a Student class. 12. Create a program to illustrate the concept of method overriding using a parent and child class. 13. Write a Java program using multiple inheritance through interfaces to implement a simple vehicle management system. 14. Develop a Java program to create a package named MyPackage and import it in another program. 15. Implement a Java program using the instanceof operator to check object types in an inheritance hierarchy. **Objective :** To apply the principles of object-oriented programming in Java by implementing concepts of inheritance, polymorphism, encapsulation, class management, interfaces, and file handling to build robust and scalable applications.	LO4
6.0	1. Write a Java program to demonstrate exception handling using try, catch, and finally blocks. 2. Develop a Java program to create a custom exception class and handle it using throw and throws keywords..	• Practical 1–5 → LO5 (Exception Handling) • Practical 6–10 → LO5 (Multithreading) • Practical 11–15 → LO5

	3. Create a Java application that handles multiple exceptions using multiple catch blocks. 4. Write a Java program to demonstrate nested try statements. 5. Implement a program to simulate uncaught exceptions and analyze its impact.. 6. Write a Java program to demonstrate thread creation using Runnable and Thread classes. 7. Develop a Java program to implement thread synchronization using synchronized methods. 8. Create a Java application to demonstrate inter-thread communication using wait(), notify(), and notifyAll(). 9. Write a Java program to demonstrate thread priorities and daemon threads.. 10. Implement a program to show the lifecycle of a thread using different states. 11. Create a simple Java applet that displays a welcome message using the paint() method 12. Develop a Java applet to handle mouse events and display coordinates. 13. Write a Java program to create a graphical user interface using AWT components. 14. Implement a Java applet that takes parameters from HTML using the Applet tag. 15. Create a simple drawing application using AWT that allows the user to draw shapes. **Objective :** To understand and implement advanced Java concepts such as exception handling, multithreading, applet programming, and graphical user interface (GUI) development using AWT, enhancing application reliability and user interactivity.	(Applets and GUI)

List of Mini project

Sr No	List of Mini Projects (CPP)	LO
01	**Student Report Card Management System** • **Concepts Used:** File Handling, Classes, Structures, Constructors • **Description:** Add, delete, modify, and view student academic records. Data is stored in files.	LO2, LO3
02	**Library Management System** • **Concepts Used:** File Handling, OOP, Arrays, Structures	LO2, LO3

	• **Description:** Manage books in a library (add/remove/search). Track issued and returned books.	
03	**Bank Management System** • **Concepts Used:** File Handling, Classes & Objects, Data Abstraction • **Description:** Simulate banking operations like opening an account, deposits, withdrawals, and account balance inquiry.	LO2, LO3
04	**Inventory Management System** • **Concepts Used:** File I/O, Object-Oriented Design, Polymorphism • **Description:** Track product stock, purchase and sales, and maintain transaction logs.	LO2, LO3
05	**Hotel Reservation System** • **Concepts Used:** File Handling, Class Inheritance, Constructors • **Description:** Manage room bookings, cancellations, and availability checks with cost estimation.	LO2, LO3
06	**Employee Payroll Management System** • **Concepts Used:** File Handling, Inheritance, Virtual Functions • **Description:** Calculate salary, taxes, bonuses, and store payroll details of employees.	LO2, LO3
07	**Clinic Patient Record System** • **Concepts Used:** File Streams, Class Hierarchy, Object Persistence • **Description:** Maintain patient records, appointment scheduling, and doctor allocation.	LO2, LO3
08	**Online Quiz Management System** • **Concepts Used:** File Handling, Menus, Functions, Classes • **Description:** Conduct multiple quizzes, store scores, and allow user login with progress tracking.	LO2, LO3
09	**Simple Railway Ticket Booking System** • **Concepts Used:** File I/O, Structures, Functions • **Description:** Simulate booking, cancellation, and displaying train details using file storage.	LO2, LO3
10	**Book Store Management System** • **Concepts Used:** File Handling, OOP Principles, Sorting and Searching • **Description:** Add/update/search/delete book details, generate billing and inventory reports.	LO2, LO3
	List of Mini Projects (JAVA)	
01	**Expense Tracker** **Problem Statement:** Build a desktop application where users can log daily expenses, categorize them (food, travel, etc.), and view monthly summaries. Data should be stored and retrieved from local files GUI Interface.	LO4, LO5
02	**Recipe Book Manager** **Problem Statement:** Create a GUI app where users can store, search, and edit their favorite recipes. Each recipe should be saved as an individual file or organized into categories using folders.	LO4, LO5

03	**Bug Tracker Tool** **Problem Statement:** Build a mini bug-tracking system where users can log, update, and mark bugs as resolved. Save bug reports to a file and display them in a sortable GUI table.	LO4, LO5
04	**Fitness and Workout Logger** **Problem Statement:** Allow users to create workout plans, log completed exercises, and track progress through charts or stats (optional). File handling should maintain history and progress logs and GUI.	LO4, LO5
05	**Event Scheduler and Reminder** **Problem Statement:** Design a scheduling system to plan events and get pop-up reminders. Save and load event lists using file handling, optionally integrating a basic calendar UI.	LO4, LO5
06	**Simple Customer Feedback Collector** **Problem Statement:** Build a feedback form where users submit their opinions on products or services. Responses should be saved in structured format (CSV/JSON) for analysis later and GUI.	LO4, LO5
07	**Contact Book with Export Feature** **Problem Statement:** Implement a GUI-based contact book that allows adding/editing/deleting contacts and exporting data to a .csv file for external use.	LO4, LO5
08	**Parking Lot Management System** **Problem Statement:** Simulate a parking lot with slots for vehicles. Allow entry/exit registration, generate parking slips, and save logs of all vehicles using file handling and GUI.	LO4, LO5
09	**Daily Mood Tracker** **Problem Statement:** Let users record their mood daily with a short note. Store entries in files and allow users to browse past entries and see frequency stats of mood types using GUI.	LO4, LO5
10	**Digital Notes Organizer** **Problem Statement:** Create an app to manage and organize personal notes. Provide options to add, delete, and edit notes, with autosave features and organized storage using file structures and GUI.	LO4, LO5

Online Repository

SrNo	Respository

1	**GitHub**
	• **Link:** https://github.com
	• **Reason to use It:**
	  ○ Largest open-source platform with thousands of C++ and Java projects.
	  ○ Great for exploring real-world applications, contributing to open source, and version control.
	  ○ Students can fork repositories, collaborate on code, and showcase projects for placements.

2	**GeeksforGeeks – Practice and Code Repository** • **Link:** https://practice.geeksforgeeks.org/ • **Reason to use It:** o Rich in C++ and Java examples, coding problems, and data structures. o Covers programming concepts with working source code. o Regularly updated with interview questions and competitive coding challenges.
3	**GitLab** • **Link:** https://gitlab.com • **Reason to use It:** o Similar to GitHub but with more private repositories (ideal for academic use). o Good for hosting collaborative coding projects and using CI/CD pipelines. o Supports C++ and Java with various development tools.
4	**SourceForge** • **Link:** https://sourceforge.net • **Reason to use It:** o Repository of open-source software including tools, utilities, and programming frameworks. o Many Java GUI-based and C++ utility projects are available for download and modification. o Ideal for exploring legacy and niche programming projects.
5	**CodeChef GitHub Repository (and Platform)** • **Link:** o CodeChef: https://www.codechef.com/ o GitHub Repo: https://github.com/codechef • **Reason to use It:** o Offers a massive problem-solving community for C++ and Java. o Students can practice competitive programming and refer to community-driven solutions. o Excellent for mastering logic and problem-solving patterns.

Term Work:

- At least **12** experiments (**06 experiments** each on **C++** and **JAVA**) covering entire syllabus should be set to have well predefined inference and conclusion. Teacher should refer the suggested experiments and can design additional experiment to maintain better understanding and quality.
- The experiments should be students centric and attempt should be made to make experiments more meaningful, interesting and innovative.
- Term work assessment must be based on the overall performance of the student with every Experiments are graded from time to time.
- The grades will be converted to marks as per "**Attendance+performance+submission+Viva/MCQ Test**" and should be added and averaged. Based on above scheme grading and term work assessment should be done.

- The practical and oral examination will be based on entire syllabus. Students are encouraged to share their experiments codes on online repository. Practical exam should cover all **12** experiments for examination.
- Mini project either in CPP or java from the topic given or any other topic of same level which should include construct of CPP/Java, File handling and GUI is mandatory.

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list and Mini project. Also, Term work Journal must include at least 4 (2 CPP+ 2Java) assignments.

Term Work Marks: 50 Marks (Total marks) = 10 (Attendance) + 10 (Performance in Lab) + 10 (Timely Submission) + 20 (Viva or MCQ Test). MCQ test can be conducted using online system for which10-15 min can be allocated in every practical.

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

AC –
Item No. –

As Per NEP 2020

University of Mumbai

Syllabus for HSSM Vertical 5		
Faculty of Engineering		
Board of Studies in Under Engineering		
Second Year Programme in HSSM– Common to All Branches		
Semester	**III & IV**	
Title of Paper (Lab)		**Credits**
I) Entrepreneurship Development	**III**	**2**
II) Environmental Science	**III**	**2**
III) Business Model Development	**IV**	**2**
IV) Design Thinking	**IV**	**2**
Total Credits		**8**
From the Academic Year		**2025-26**

Sem. - III

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2993511	Entrepreneurship Development	--	2*+2	-	-	2*+2	-	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2993511	Entrepreneurship Development	--	--	--	--	50	--	50

Note: * Two hours of practical class to be conducted for full class as demo/discussion/theory.

Lab Objectives:

1. To introduce students to entrepreneurship concepts and startup development.
2. To develop business idea generation, validation, and business model preparation.
3. To provide hands-on experience in market research, financial planning, and business pitching.
4. To enhance problem-solving and decision-making skills in entrepreneurial ventures.
5. To familiarize students with government schemes and support systems for entrepreneurs.
6. To develop communication and presentation skills required for business pitching.

Lab Outcomes:

Upon successful completion of this course, students will be able to:

1. Understand the fundamental concepts of entrepreneurship and business models.
2. Conduct market research and develop business plans.
3. Utilize financial planning and cost analysis for startups.
4. Apply entrepreneurial skills to identify and solve business challenges.
5. Develop prototypes using open-source software for business operations.
6. Pitch business ideas effectively with structured presentations.

DETAILED SYLLABUS

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	**Prerequisite**	Fundamentals of communication and leadership skills.	01	--
I	**Introduction to Entrepreneurship**	Definition, Characteristics, and Types of Entrepreneurs. Entrepreneurial Motivation and Traits. Start-up Ecosystem in India. Challenges in Entrepreneurship	**02**	**LO1**
II	**Business Idea Generation &**	Ideation Techniques: Design Thinking, Brainstorming, Mind	**04**	**LO2**

	Validation	Mapping. Business Model Canvas (BMC). Market Research & Customer Validation. Minimum Viable Product (MVP) Concept.		
III	**Business Planning & Strategy**	Writing a Business Plan. SWOT Analysis and Competitive Analysis. Financial Planning and Budgeting. Risk Assessment and Management	**04**	**LO3**
IV	**Funding and Legal Framework**	Sources of Funding: Bootstrapping, Angel Investors, Venture Capital Government Schemes & Start-up India Initiatives. Business Registration & Legal Formalities. Intellectual Property Rights (IPR) & Patents	**05**	**LO4**
V	**Marketing & Digital Presence**	Branding and Digital Marketing. Social Media Marketing & SEO. Customer Relationship Management (CRM). E-commerce & Online Business Models	**05**	**LO5**
VI	**Business Pitching & Prototype Development**	Pitch Deck Preparation & Presentation Techniques. Prototyping with Open-source Tools. Elevator Pitch & Investor Pitch. Case Studies of Successful Start-ups	**05**	**LO6**

Text Books:

1. "Entrepreneurship Development and Small Business Enterprises" – Poornima M. Charantimath, Pearson, 3rd Edition, 2021.
2. "Innovation and Entrepreneurship" – Peter F. Drucker, Harper Business, Reprint Edition, 2019.
3. "Startup and Entrepreneurship: A Practical Guide" – Rajeev Roy, Oxford University Press, 2022.
4. "Essentials of Entrepreneurship and Small Business Management" – Norman Scarborough, Pearson, 9th Edition, 2021.
5. "The Lean Startup" – Eric Ries, Crown Publishing, 2018.

References:

1. "Disciplined Entrepreneurship: 24 Steps to a Successful Startup" – Bill Aulet, MIT Press, 2017.
2. "Zero to One: Notes on Startups, or How to Build the Future" – Peter Thiel, 2014.
3. "The $100 Startup" – Chris Guillebeau, Crown Business, 2019.
4. "Business Model Generation" – Alexander Osterwalder & Yves Pigneur, Wiley, 2020.
5. "Blue Ocean Strategy" – W. Chan Kim & Renée Mauborgne, Harvard Business Review Press, 2019.

Online Resources:

Website Name
1. Startup India Portal – https://www.startupindia.gov.in 2. MIT OpenCourseWare – Entrepreneurship – https://ocw.mit.edu/courses/sloan-school-of-management/ 3. Coursera – Entrepreneurship Specialization – https://www.coursera.org/specializations/entrepreneurship

4. Harvard Business Review – Entrepreneurship Articles – https://hbr.org/topic/entrepreneurship
5. Udemy – Startup & Business Courses – https://www.udemy.com/courses/business/entrepreneurship/

List of Experiments.

Sr No	List of Experiments	Hrs
01	Business Idea Generation using Mind Mapping.	02
02	Conducting Market Research & Customer Validation.	02
03	Preparing a Business Model Canvas for a Startup Idea.	02
04	Developing a Financial Plan & Break-even Analysis.	02
05	Creating a Website using WordPress/Wix.	02
06	Social Media Marketing Campaign using Open-source Tools.	02
07	Digital Prototyping using Figma/Inkscape.	02
08	Business Pitch Deck Preparation & Presentation.	02
09	Exploring Government Schemes for Startups.	02
10	Legal Compliance & IPR Basics (Case Study).	02

Sr No	List of Assignments / Tutorials	Hrs
01	a. Write a report on any successful entrepreneur and their startup journey. b. Conduct SWOT analysis for a real-life startup.	02
02	Develop a business idea and create a one-page business plan.	02
03	Conduct market research using surveys & present findings.	02
04	Design a simple logo and branding strategy for a startup.	02
05	Create a financial model and cost estimation for a startup.	02
06	Make a case study report on startup failure analysis.	02

List of Open-Source Software
1. Canva – Designing pitch decks, social media posts, and branding materials. 2. Trello / Asana – Project management for startups. 3. GIMP / Inkscape – Graphic design and logo creation. 4. WordPress / Wix – Website development for startups. 5. OpenCart / PrestaShop – E-commerce website setup. 6. Figma – UI/UX design and prototyping. 7. LibreOffice Calc – Financial planning and budgeting. 8. Google Suite (Docs, Sheets, Slides) – Documentation and presentations. 9. Python (Pandas, Flask, Django) – Data analytics and web application development. 10. MailChimp – Email marketing and customer engagement.

Assessment :

Term Work: Term Work shall consist of at least 08 to 10 practicals' based on the above list. Also, Term work Journal must include at least 6 assignments.

Term Work Marks: 50 Marks (Total marks) = 20 Marks (Experiment) + 15 Marks (Assignments) + 5 Marks (Attendance)+ 10 Marks (Report)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2993512	Environmental Science	--	2*+2	-	--	2*+2	-	2

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I+IAT-II					
2993512	Environmental Science	--	--	--	--	--	50	--	50

Note: * Two hours of practical class to be conducted for full class as demo/discussion/theory.

Rationale:

Most of the engineering branches are offspring of applied sciences, and their practices have a significant impact on the environment. Understanding environmental studies is essential for engineers to develop sustainable solutions, minimize ecological footprints, and promote responsible resource management. This course equips students with the knowledge of ecosystems, biodiversity, pollution control, and environmental laws, enabling them to integrate sustainability into engineering practices.

Lab Objectives:

1. To understand the scope, importance, and role of environmental studies in public awareness and health.
2. To study different natural resources, their issues, and sustainable conservation.
3. To understand ecosystem types, structures, and functions.
4. To explore biodiversity, its importance, threats, and conservation.
5. To learn about pollution types, causes, effects, and control measures.
6. To understand environmental challenges, sustainability, and ethics.

Lab Outcomes:

1. Explain the significance of environmental studies and the role of IT in environment and health.
2. Describe resource types, associated problems, and conservation methods.
3. Classify ecosystems and explain their role in ecological balance
4. Analyze biodiversity levels and conservation strategies, especially in India.
5. Explain pollution impacts and suggest preventive measures.
6. Discuss environmental issues and propose sustainable solutions.

DETAILED SYLLABUS:

Unit Name	Topic Name	Topic Description	Hours	LO Mapping

I	**The Multidisciplinary Nature of Environmental Studies**	Definition, scope and importance. Need for public awareness, Role of information technology in environment and human health. Human population and the environment: Population growth, variation among nations. Population Explosion- family welfare program. Environment and human health Women and child welfare	**03**	LO1
II	**Natural Resources**	Renewable and non-renewable resources. Natural resources & associated problems: a) Forest resources: b) Water resources: Natural resources & associated problems c) Mineral resources: d) Food resources: e) Energy resources: Role of an individual in conservation of natural resources: f) Equitable use of resources for sustainable lifestyles.	**04**	LO2
III	**Ecosystems**	Concepts of an ecosystem. Introduction, types, characteristic features, structure and function of the following ecosystem: a. Forest ecosystem b. Grassland ecosystem c. Desert ecosystem d. Aquatic ecosystem (ponds, streams, lakes, rivers, oceans, estuaries). Case study on various ecosystems in India.	**05**	LO3
IV	**Biodiversity and its Conservation**	Introduction-Definition: genetic species and ecosystem diversity. Bio-geographical classification of India Value of biodiversity : Consumptive use, productive use, social, ethical, aesthetic and option values, Bio-diversity at global, national, local levels India as a mega diversity nation Case study on Bio diversity in India.	**05**	LO4
V	**Environmental Pollution Definition**	Causes, effects and control measures of: a) Air pollution b) Water pollution c) Soil pollution. Solid waste management: Causes, effect and control measures of urban and industrial wastes. Role of an individual in prevention of pollution, Case study on Pollution Disaster management: floods, earthquake, cyclone and landslides. Carbon Credits for pollution prevention	**05**	LO5

VI	**Social Issues and Environment**	From unsustainable to sustainable development Urban problems related to energy, Water conservation, rain water harvesting, watershed management. Environmental ethics: issues and possible solution. Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case studies. Consumerism and waste products. Environment protection act. Public awareness Case study on Environmental Ethics	**04**	LO6

Textbooks

1. Environmental Science: Towards a Sustainable Future, G. Tyler Miller and Scott Spoolman, 13th Edition, Cengage Learning 2021
2. Environmental Management: Text and Cases, Bala Krishnamoorthy, 3rd Edition, PHI Learning, Publication Year: 2016
3. Green IT: Concepts, Technologies, and Best Practices, Markus Allemann, Springer 2008
4. Sustainable IT: Slimming Down and Greening Up Your IT Infrastructure, David F. Linthicum, IBM Press 2009
5. Environmental Modelling: Finding Solutions to Environmental Problems, David L. Murray, Cambridge University Press 2016
6. Remote Sensing and Image Interpretation, Thomas M. Lillesand, Ralph W. Kiefer, and Jonathan W. Chipman, 9th Edition, John Wiley & Sons 2020
7. Business Ethics: Concepts and Cases, Manuel Velasquez, Pearson 2012

Reference Books

1. Environmental Law and Policy in India, Shyam Divan and Armin Rosencranz, 2nd Edition, Oxford University Press 2018
2. The International Handbook of Environmental Laws, David Freestone and Jonathon L. Rubin, Edward Elgar Publishing 2000
3. E-Waste Management: Challenges and Opportunities in Developing Countries, Ruediger Kuehr and Ram K. Jain, Springer 2008
4. The E-Waste Handbook: Managing Electronic Waste, Klaus Hieronymi, Ruediger Kuehr, and Ram K. Jain, Earthscan 2009
5. Environmental Ethics: An Introduction, J. Baird Callicott, University of Georgia Press1999

Online References:

Sr. No.	Website Name
1.	Centre for Science and Environment (CSE), Website: cseindia.org
2.	Ministry of Environment, Forest and Climate Change (MoEFCC), Government of India
3.	CSIR-National Environmental Engineering Research Institute (NEERI)

List of Experiments.

Sr No	List of Experiments	Hrs

01	Study of Environmental Components and Ecosystems.	2
02	Visit and Report on Solid Waste Management Plant.	2
03	Study of Renewable Energy Sources (Solar, Wind, Biogas).	2
04	Analysis of Air and Water Quality Parameters.	2
05	Study of Local Biodiversity and Conservation Methods.	2
06	Awareness Activity on Environmental Issues.	2
07	Rainwater Harvesting System Design	2
08	Case Study on Environmental Pollution & Control Measures.	2
09	Report on Climate Change Impact and Adaptation.	2
10	Study of Environmental Laws and Acts.	2
11	Study of Disaster Management Techniques.	2
12	Report on Role of IT in Environmental Protection.	2

Sr No	List of Assignments / Tutorials	Hrs
01	Prepare a report on Renewable and Non-Renewable Resources.	2
02	Write a case study on Ecosystem Types in India	2
03	Write a report on Biodiversity in India.	2
04	Prepare a report on Pollution Types and Control Measures.	2
05	Prepare a report on Environmental Ethics and Sustainability.	2
06	Prepare a case study report on Global Warming and Climate Change.	2
07	Report on Role of an Individual in Environmental Protection.	2
08	Write a report on Disaster Management Techniques.	2
09	Prepare a report on Environmental Laws and Acts in India.	2
10	Case Study on E-waste Management and Recycling Techniques.	2

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 8 to 10 assignments.

Term Work Marks: 50 Marks (Total marks) = 20 Marks (Experiment) + 15 Marks (Assignments) + 5 Marks (Attendance)+ 10 Marks (Report)

Vertical – 6

Experiential Learning Courses

(CEP)

Detailed Syllabus

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2303611	Smart Embedded Systems with Atmega 168/328	-	4	-	-	2	-	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		Test 1	Test 2	Avg. of 2 Tests				
2303611	Smart Embedded Systems with Atmega 168/328	--	--	--	--	50	25	75

Prerequisite:

- C++ and Java Programming
- Electronic Devices and Circuit
- Digital System Design

Course Objectives:

1. To understand the architecture, memory organization, and peripheral features of the ATmega168/328 microcontroller.
2. To explore internal functional blocks like timers, ADC, PWM, and serial communication interfaces.
3. To gain proficiency in embedded C programming using development tools for microcontroller programming and interfacing.
4. To interface wireless modules, sensors, actuators, and integrate IoT platforms for real-time control and monitoring.
5. To design, develop, and convert a prototype into a marketable embedded product considering DFM and cost analysis.
6. To design schematics and PCB layouts using EDA tools and apply fabrication standards

Course Outcomes:

After successful completion of the course student will be able to:

1. Identify pin configurations, explain memory types, and write simple code to configure I/O ports.
2. Configure and program internal peripherals like timers, interrupts, and serial communication modules (USART, SPI, I2C).
3. Develop, compile, and upload programs using IDEs, and implement sensor interfacing and interrupt handling.

4. Design and build embedded applications integrating sensors, actuators, displays, and wireless modules with cloud-based data logging and control.
5. Create a prototype, optimize it for production, and present a final project report including BOM and feasibility study.
6. Create a PCB layout, follow design rules, and generate Gerber files for the fabrication of microcontroller-based projects

DETAILED SYLLABUS:

Sr No	Detailed Content	Hours	LO Mapping
I	**Introduction to ATmega168/328 Microcontroller**	**05**	LO1
	Focus: This unit introduces the fundamental concepts of **microcontrollers**, their difference from microprocessors, and specifically focuses on the **AVR family** with an emphasis on **ATmega168/328**. It covers **architecture, pin configuration, memory organization, and peripheral features**, helping students understand how to work with these controllers in embedded systems.		
	Key Skills: • Understanding microcontroller architecture • Pin Mapping and Peripheral Understanding • Memory Organization & Power Management • Programming I/O Ports • Clock and Power Management		
	Syllabus: • Overview of Microcontrollers and Microprocessors • Introduction to AVR Family • Features and Specifications of ATmega168/328 • Pin Diagram and Pin Description • Clocking and Power Management • Memory Organization: Flash, SRAM, EEPROM • I/O Ports and Peripheral Features		
	Activities: • Study and analyze the **ATmega168/328** datasheet • Identify and explain **pin configurations** and their functions • Discuss **Flash, SRAM, EEPROM**, and clocking system • Write simple **code to configure and control I/O ports** • Explore different **clock sources and power-saving modes**		
	Self-Learning:		
	• RAM and its Types of RAM • ROM and Its Types • Difference between RAM and ROM • Different ways of Clock Generation		

II	ATmega168/328 Architecture	07	LO2
	Focus: This unit explores the **internal architecture and functionality** of the **ATmega168/328 microcontroller**, covering essential components such as registers, timers, interrupts, ADC, PWM, and communication protocols (USART, SPI, and I2C). The primary objective is to help students understand how the microcontroller processes data, interacts with external components, and handles real-time operations.		
	Key Skills:		
	• Understanding the ATmega168/328 Architecture • Register and Memory Utilization • Timers and Counters • Interrupt Handling • ADC (Analog to Digital Conversion) • PWM (Pulse Width Modulation) • USART, SPI, and I2C Communication		
	Syllabus: • Detailed Architecture and Block Diagram • Registers and Memory Mapping • Timers and Counters • Interrupt Handling • ADC (Analog to Digital Conversion) • PWM (Pulse Width Modulation) USART, SPI, and I2C Communication		
	Activities: • Study the **block diagram** and analyze internal components. • **Hands-on coding** for configuring **registers and memory-mapped I/O**. • Write a **program to implement delay generation** using timers. • Configure and test **external and internal interrupts** using ISR functions. • Interface a **potentiometer or sensor** and read analog values. • Generate **PWM signals** to control LED brightness or motor speed. • Establish serial communication between **ATmega168/328 and peripherals** (e.g., sensors, LCD, Bluetooth module).		
	Self-Learning:		
	• Types of IO Mapping • Interrupt and its types • PWM (Pulse Width Modulation) • Types of Communication (serial and Parallel) • Difference between Serial and Parallel Communication		

	• Serial Communication Frame format(UART)		
III	**Programming ATmega168/328**	**10**	LO3
	Focus: This unit introduces **Embedded C programming** and the necessary **software tools (IDEs)** for programming microcontrollers. It focuses on **basic coding, hardware interfacing**, and **handling interrupts**, providing students with the fundamental skills required for embedded system development.		
	Key Skills: • Understanding Embedded C Programming • Installing and Setting Up IDEs • Writing and Compiling Code in Arduino IDE • GPIO (General Purpose Input/Output) Interfacing • Sensor Interfacing • Serial Communication • Interrupt Handling		
	Syllabus: • Introduction to Embedded C Programming • Basic Data types, Functions & Pointers, Data Structures, • Memory & Optimization, Bit Manipulation & Efficiency. • Installation and Setup of IDEs like Keil, Atmel Studio / Microchip Studio, PlatformIO (with VS Code), Eclipse with AVR Plugin (AVR-Eclipse),AVR-GCC + Makefiles (Bare Metal CLI) • GPIO Interfacing and LED Blinking • Analog and Digital Sensor Interfacing • Serial Communication Programming • ISR for Interrupt handling		
	Activities: • **Write and execute basic C programs** for microcontrollers. • Install **Atmel Studio, or PlatformIO** and configure settings. • Develop **basic programs**, compile and upload them to a microcontroller. • Implement **LED blinking** and control output pins using code. • Interface **analog (e.g., temperature sensor) and digital sensors (e.g., IR sensor)**. • Implement **UART communication** to send and receive data. • Use **Interrupt Service Routines (ISRs)** to handle external events (e.g., button press detection).		
	Self-Learning:		
	• Basics of embedded c programming • Various IDE for microcontroller programming • I/O pins and sensor interfacing • Serial communication and interrupt handling		

Unit 4	**Advanced Technologies and Project Integration**	**18**	LO4
	Focus: This unit focus **wireless communication, sensor and actuator interfacing, display modules, motor control, and IoT integration** with **ATmega microcontrollers**. The goal is to develop skills in **real-time data acquisition, processing, and remote monitoring/control** using cloud platforms.		
	Key skills: • **Wireless Communication** (Bluetooth, Wi-Fi, Zigbee, RF 434MHz, LoRa) • **Sensor Interfacing** (Gas, Fire, DHT11, Obstacle, Soil Moisture, etc.) • **Actuator Interfacing** (Relay, Motor, Buzzer) • **Display Interfacing** (LCD, LED, OLED, 7-Segment) • **Motor Control with L298N Motor Driver** • **IoT Integration with ATmega** (ThingSpeak, MQTT, Blynk,Google Firebase) • **Real-Time Data Monitoring and Control**		
	Syllabus: • Introduction to Wireless Communication (Bluetooth, Wi-Fi, Zigbee, RF434Mhz, LoRa) • Introduction to modules of Wireless identification (RFID-RC-522, EM-18, nRF24L01) • Introduction to Sensors like IR (Obstacle, Gas, Fire, DHT11, Obstacle, Soil Moisture, Ultrasonic(HC-SR-04), Touch, Biometric Pressure(BMP180/280),Soil Sensor moisture ,water Sensor, Heart Rate (Max30100),PIR, Gyroscope(MPU6050), magneto meter(HMC5883L), Accelerometer(ADXL345), Finger Print Sensor(R305), Camera Modules (e.g., OV7670), Strain Gauge, Voltage(ZMPT101B), Current(ACS712) and Power Sensor(PZEM-004T) or similar etc. • Introduction to Actuator like Relay, Motor (DC Motor, Stepper, Servo), Buzzer • Motor Control using L293D/L298N Motor Driver • Relay Control using ULN2803 • Stepper Control using ULN 2003/2803 • Servo Motor Control • Interfacing of LCD, LED, OLED,7Segment • IoT Integration with ATmega (Cloud Like Thingspeak, MQTT, Blynk, Google Firebase or Similar) • Real-Time Data Monitoring and Control • Using above Hardware/Controller define the Project define problem statement to make a project and into product		

	Activities: • Establish wireless communication between **microcontroller and other devices** for data exchange. Example: **Control LEDs using a Bluetooth app**. • Connect sensors to **ATmega** and write code to **collect and process sensor data**. Example: **Display temperature readings from DHT11**. • Use **relays and buzzers** for control actions. Example: **Activate a buzzer when gas is detected**. • Display sensor readings and system status on **LCD/OLED**. Example: **Show soil moisture levels on OLED display**. • Control **DC motors and stepper motors** using **L298N motor driver**. Example: **Move a robotic car forward and backward**. • Connect microcontroller to **cloud platforms** for **real-time monitoring and control**. Example: **Upload temperature data to ThingSpeak** and view it on a dashboard. • Develop a system to **monitor and control devices remotely**. Example: **Send sensor data to a web dashboard and control relays from a mobile app**		
	Self-Learning:		
	• Wireless communication and its type • Sensor and its pin diagram for interfacing • Actuator and its pin diagram for interfacing • Display and its pin diagram for interfacing • IOT integration with embedded system		
V	**Project Development, Testing and Product Conversion**	**4+4**	LO5
	Focus:		
	This unit focuses on **end-to-end product development**, from **conceptualization to market-ready product conversion**. It covers **system design, prototyping, testing, debugging, hardware-software integration, and final manufacturing considerations**.		
	Key Skills: • **Design for Manufacturing (DFM)** • **Enclosure Design** • **Cost Estimation & Bill of Materials (BOM)** • **Market Analysis & Feasibility Study** • **Final Project Report & Presentation**		
	Syllabus (Part-1): • System Design and Block Diagram Creation • Component Selection and Circuit Design • Prototype Development and Testing • Debugging Techniques • Software and Hardware Integration		

	Syllabus (Part-2): • Design for Manufacturing (DFM) • Enclosure Design • Cost Estimation and Bill of Materials (BOM) • Market Analysis and Feasibility Study • Final Project Report, Presentation and Research Paper		
	Activities: • Optimize the design for **mass production**. Example: **Ensure PCB layout follows industry-standard manufacturing constraints**. • Develop **3D-printed or injection-molded enclosures** for the product. Example: **Design a waterproof casing for an outdoor sensor**. • Analyze the **cost-effectiveness** of components. Example: **Prepare a BOM with pricing details for a consumer IoT device**. • Conduct a study on **potential users and competitors**. Example: **Analyze demand for a smart wearable device**. • Document findings and present the **prototype-to-product journey**. Example: **Showcase a working prototype with performance insights**.		
	Self-Learning:		
	• 3D Printing, Types of 3D printing, • Material used in 3D printing • Basics of 3D printing model • 3D modelling Software, how to use that software • Basic knowledge of cost estimation and bills of material • Idea to product development procedure		
VI	**PCB Design for Microcontroller Projects**	**04**	LO6
	Focus: This unit introduces **Printed Circuit Board (PCB) design**, covering **schematic creation, layout design, tools, design rules, and fabrication processes**. The goal is to develop skills for designing efficient and manufactural PCBs using industry-standard software		
	Key Skills: • **Understanding PCB Design** • **Schematic Creation & PCB Layout Design** • **Using PCB Design Software** (Eagle, Altium, KiCAD, EasyEDA) • **Design Rules & Guidelines** • **PCB Fabrication Process**		
	Activities: • Learn **PCB design fundamentals**, importance, and real-world applications. Example: **Exploring single-layer and multi-layer PCB designs**.		

	• Design circuit schematics and convert them into PCB layouts. Example: **Create a simple LED driver circuit in Proteus/Eagle**. • Work with **PCB design tools** to simulate and design circuits. Example: **Simulate an Arduino-based circuit in Proteus**. • Learn **trace width, spacing, ground planes, and via placement** for optimized PCB design. Example: **Follow proper clearance rules to prevent short circuits**. • Understand **etching, drilling, solder masking, and component placement** for PCB manufacturing. Example: **Design a PCB and generate Gerber files for fabrication for the project taken by students**		
	Syllabus: • Introduction to PCB Design • PCB Layout and Schematic Creation • Tools for PCB Design: Proteus, Eagle, Altium Designer, KiCAD, Easy EDA or any similar tool • Design Rules and Guidelines • PCB Fabrication Process		
	Self-Learning:		
	• PCB, Types of PCB • Material used for PCB and their advantages, Disadvantages and Application • Standard PCB design processes • Basic knowledge about PCB design tools • PCB design rules for compact design structure • PCB fabrication process		

Textbooks:

1. "AVR Microcontroller and Embedded Systems" by Muhammad Ali Mazidi, Sarmad Naimi, and Sepehr Naimi.
2. "Programming and Customizing the AVR Microcontroller" by Dhananjay V. Gadre.
3. "Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers" by Jonathan Valvano.
4. "Introduction to Embedded Systems" by Shibu K V.
5. "Embedded C Programming and the Atmel AVR" by Barnett, Cox, and O'Cull.
6. **Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing" by** Ian Gibson, David Rosen, Brent Stucker
7. **PCB Design Tutorial" by** David L. Jones (EEVblog)

Reference Books:

1. "AVR Programming: Learning to Write Software for Hardware" by Elliot Williams
2. "Microcontroller Projects in C for the 8051" by Dogan Ibrahim

3. "The AVR Microcontroller and Embedded Systems Using Assembly and C" by Muhammad Ali Mazidi.
4. "Designing Embedded Systems with PIC Microcontrollers" by Tim Wilmshurst.
5. "Fundamentals of Microcontrollers and Applications in Embedded Systems" by Ramesh S. Gaonkar
6. **"3D Printing for Dummies"** *(Introductory & DIY-friendly)* **By** Kalani Kirk Hausman, Richard Horne
7. **"Designing Printed Circuits Boards with KiCad" by** Peter Dalmaris

Software Tools:

1. **Arduino IDE: https://www.arduino.cc/en/software**
2. **Atmel Studio: https://www.microchip.com/en-us/tools-resources/develop/microchip-studio**
3. **Microchip Studio: https://www.microchip.com/en-us/tools-resources/develop/microchip-studio**
4. **PlatformIO: https://platformio.org/install/integration**

   https://platformio.org/platformio-ide
5. **Visual Studio Code: https://code.visualstudio.com/**
6. **AVR-GCC (GNU AVR C Compiler):**

   https://gcc.gnu.org/wiki/avr-gcc
7. **SimulAVR or AVRSim (Simulation Tools): https://sourceforge.net/projects/simulavr/**
8. **3D Model Design Website**

   https://www.sketchup.com/

   Spline - 3D Design tool in the browser with real-time collaboration

   3D Models for Free - Free3D.com

Online Resources:

Sr No	Website
1	Microchip Technology – ATmega328P Datasheet: Detailed specifications and features of the ATmega328P microcontroller.
	https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
2	AVR Libc – NonGNU.org: Comprehensive user manual for AVR Libc, providing standard C library functions for AVR microcontrollers.nongnu.org+1Courses at Washington University+1
	https://www.nongnu.org/avr-libc/user-manual/

3.	Arduino Official Website: Resources, tutorials, and documentation for Arduino boards and software.
	https://www.arduino.cc/
4.	SparkFun Inventor's Kit Experiment Guide: Step-by-step projects and experiments using the SparkFun Inventor's Kit.Microchip Docs+2SparkFun Learn+2SparkFun Learn+2
	https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v41
5	Microchip Developer Help: Comprehensive online resource for developers using Microchip's products and technologies.Developer Help
	https://developerhelp.microchip.com/xwiki/bin/view/Main/
6	KiCad EDA – Schematic Capture & PCB Design Software: Open-source software suite for electronic design automation.Altium+6Wikipedia+6KiCad EDA+6
	https://www.kicad.org/
7	Autodesk EAGLE: Electronic design automation software for PCB designers.Autodesk
	https://www.autodesk.com/products/eagle/overview
8	Altium Designer: Industry-leading PCB design software combining schematic, layout, and more.Altium
	https://www.altium.com/
9	ThingSpeak – IoT Analytics Platform: Platform for aggregating, visualizing, and analyzing live data streams in the cloud.Halvorsen Blog+3ThingSpeak+3MathWorks - Maker of MATLAB and Simulink+3
	https://thingspeak.com/
10	Blynk IoT Platform: Low-code platform for building and managing IoT applications.
	https://blynk.io/
11	3D printing : Tinkercad (by Autodesk)
	https://www.tinkercad.com
12	3D printing : FreeCAD (Open Source CAD Software)
	https://www.freecadweb.org

List of Experiments:

Sr No	List of Experiments	Hours
01	Installation of IDE used and How write program, compile, and Upload	2
02	Interfacing of Buzzer, LED, Relay, Motors	4
03	Interfacing of different Sensors	4
04	Interfacing of Different cloud Platform & Database	4
05	Designing of circuit diagram of project	2
06	Designing of PCB on Software	2
07	Preparing PCB and Testing	2

08	Assembling project and Testing	4
09	Designing of enclosure on 3d Printing Software	2
10	Printing the enclosure	
11	Preparing Bill of Material, Market analysis and Feasibility study	2
12	Writing Research Paper	4

List of Mini Projects:

Note: This is tentative list student select any other good topic which should have 2-3 sensor, 2-3 actuator, 2 Cloud Platform and database.

Mini Project Topic	
1	**Title: Smart Environment Monitor: Temperature & Gas Alert System** **Synopsis:** • This project integrates a ATMEGA168/328 with a DHT11 sensor to measure temperature and humidity, along with a gas sensor to detect harmful gas leaks. • An OLED display shows real-time data locally. The system sends sensor data to both Blynk and Thingspeak for remote monitoring. • Alerts are triggered if the values exceed safe limits, ensuring prompt action. This IoT solution is ideal for home or industrial environmental safety **Sensors used**: DHT11, Gas Sensor **Actuator used:** OLED, Buzzer,LED **Cloud Platform:** Bynk, Thingspeak
2	**Title: Smart Irrigation System: Automated Plant Watering** **Synopsis:** • Using a soil moisture sensor interfaced with **ATMEGA168/328**, this project continuously monitors the moisture level of the soil. • When moisture falls below a preset threshold, a **water pump** is automatically activated to irrigate the plants. • Data is logged and displayed remotely via **Thingspeak** and **Blynk**, allowing users to track plant health. It's a practical application of IoT in agriculture and home gardening. **Sensors used:** Soil Moisture Sensor **Actuator used:** Water Pump, Buzzer, LED **Cloud Platform:** Blynk, Thingspeak
3	**Title: Dual Sensor Alert: Obstacle & Metal Detection System** **Synopsis:** • This project combines a **proximity sensor** with an **IR sensor** on an **ATMEGA168/328** platform to detect obstacles and metallic objects. • When an object is detected, the system immediately sends a signal via the **Blynk app** for

	real-time monitoring. • It can be used for security or automated machinery, ensuring safety and operational efficiency. **Sensors used:** Proximity Sensor, IR Sensor **Actuator used:** Buzzer, LED **Cloud Platform:** Blynk
4.	**Synopsis:** • Integrating an **IR sensor** with a **servo motor**, this **ATMEGA168/328** project automates parking space management. • The IR sensor detects the presence of a vehicle, while the servo motor operates the parking gate. • Communication through **MQTT** ensures real-time updates and remote monitoring of parking slot availability. **Sensors used:** IR Sensor **Actuator used:** Servo Motor, Buzzer, LED **Cloud Platform:** MQTT
5	**Title: Clean Air Sentinel: Air Quality Monitoring & Alert System** **Synopsis:** • This project uses **ATMEGA168/328** paired with an **MQ135 sensor** and an additional gas sensor to monitor ambient air quality. • Readings are displayed on an **LCD** and transmitted via **MQTT** for remote monitoring. • When pollutant levels exceed safe thresholds, the system generates an alert. It is suitable for both indoor and outdoor environments, promoting health and safety. **Sensors used:** MQ135, Gas Sensor **Actuator used:** LCD, Buzzer,LED **Cloud Platform:** MQTT, Blynk
6	**Title: Smart Water Tank: Automated Water Level Control System** **Synopsis:** • In this project, an **ultrasonic sensor** connected to **ATMEGA168/328** monitors the water level in a tank continuously. • When water levels drop below a predetermined level, a **water pump** is automatically activated to refill the tank. • Real-time data is sent through **MQTT** to a cloud-based dashboard for remote supervision. This system ensures efficient water management and minimizes wastage. **Sensors used:** Ultrasonic Sensor **Actuator used:** Water Pump, LED, Buzzer **Cloud Platform:** MQTT, Blynk
7	**Title: Smart Waste Manager: Automated Dustbin Monitoring System** **Synopsis:** • This project employs an **ultrasonic sensor** to measure the fill level of a dustbin, with a **servo**

	motor mechanism for automated lid control. • A **buzzer** sounds an alert when the bin nears capacity. Sensor data is sent to both **Thingspeak** and **Blynk**, enabling remote monitoring and timely waste collection. • The system aims to maintain cleanliness and optimize waste management processes. **Sensors used:** Ultrasonic Sensor **Actuator used:** Servo Motor, Buzzer, LED **Cloud Platform:** Thingspeak, Blynk
8	**Title: Safety First: Temperature & Fire Early Warning System** **Synopsis:** • A **DS18B20 temperature sensor** and a **fire sensor** work with **ATMEGA168/328** to monitor critical environmental parameters. • An **LCD** provides local readouts while data is sent via **MQTT** and **ThingSpeak** to cloud platforms for remote access. • Alerts are triggered when temperature or fire hazards are detected, ensuring early intervention. **Sensors used:** DS18B20, Fire Sensor **Actuator used:** LCD, Buzzer,LED **Cloud Platform:** Thingspeak, MQT
9	**Title: CloudGuard: Comprehensive Fire & Environmental Monitoring System** **Synopsis:** • This project combines **DHT11**, **gas**, and **fire sensors** with **ATMEGA168/328** to continuously monitor temperature, humidity, and fire risks. • An **LCD** displays current readings while data is uploaded to **Thingspeak** and **Blynk** for remote access. • Automated alerts notify users of any hazardous conditions, facilitating quick responses. **Sensors used:** DHT11, Gas Sensor, Fire Sensor **Actuator used:** LCD, Buzzer,LED **Cloud Platform:** Thingspeak, Blynk
10	**Title: Guardian Fall Alert: IoT-based Fall Detection and Response** **Synopsis:** • Using a **gyroscope sensor** interfaced with **ATMEGA168/328**, this system detects abnormal movements indicative of a fall. • A **buzzer** provides immediate local alerts, while sensor data is shared with **Thingspeak** and **Blynk** for remote monitoring. • The project is designed to assist elderly individuals or those at risk by ensuring rapid response in emergencies. **Sensors used:** Gyroscope Sensor,SOS button **Actuator used:** Buzzer,LED **Cloud Platform:** Thingspeak, Blynk

11	**Title: Local IoT Controller: Web-based Device Management System** **Synopsis:** • This project connects **ATMEGA168/328** to a **relay**, **buzzer**, and **light sensor** to manage multiple devices via a **local web server**. • Users can control appliances, receive feedback from the light sensor, and get alert notifications through a web interface. • It eliminates the need for external servers, ensuring quick response times within a local network. Ideal for smart home applications requiring robust local control. **Sensors used:** Light Sensor **Actuator used:** Relay, Buzzer,LED **Cloud Platform:** Local Web Server
12	**Title: Atmospheric Insights: Multi-parameter Environmental Monitor** **Synopsis:** • Utilizing the **BMP280 sensor** with **ATMEGA168/328**, this project measures temperature, pressure, and humidity simultaneously. • An **OLED display** presents real-time environmental data, while **MQTT** transmits the readings to remote platforms. • The system provides comprehensive weather insights and alerts if values deviate from normal, applicable in both indoor and outdoor environments. **Sensors used:** BMP280 **Actuator used:** OLED,Buzzer,LED **Cloud Platform:** MQTT,Blynk
13	**Title: Wireless Switchboard: Remote Device Control using RF Communication** **Synopsis:** • This project uses an **RF434 MHz wireless module** with **ATMEGA168/328** to remotely control devices via a **relay**. • It demonstrates basic wireless communication by sending on/off commands over a distance. • Useful for home automation where remote appliance control is needed, offering a simple and effective RF-based IoT solution. **Sensors used:** — **Actuator used:** Relay **Cloud Platform:** —
14	**Title: Smart Crossing Controller: Integrated Road and Railway Signal System** **Synopsis:** • In this project, **ATMEGA168/328** is connected to four **obstacle sensors** to control traffic signals at a road-railway crossing. • Sensors detect approaching vehicles or trains, triggering appropriate signal changes and automated barrier controls. • This system enhances safety and reduces the risk of accidents at high-risk intersections. **Sensors used:** Obstacle Sensor (×4)

	Actuator used: Traffic Signal LEDs, Servo Motor (if used for barrier) **Cloud Platform:** Bynk, MQTT
15	**Title: QR-Controlled Smart Home: Automated Appliance Management** **Synopsis:** • This project employs **ATMEGA168/328** to control appliances (lights, fans, bulbs) via **QR code scanning**. • By scanning a QR code, users send commands to the controller which toggles the connected devices. • Combines traditional QR technology with IoT to provide secure and convenient home automation. **Sensors used:** QR Scanner **Actuator used:** Relay (for light, fan, bulb control), Buzzer, LED **Cloud Platform:** Blynk
16	**Title: Smart Attendance Tracker: RFID-based Monitoring System** **Synopsis:** • Using an **RFID RC522 module** with **ATMEGA168/328**, this project automates attendance for classrooms or offices. • When a tag is scanned, entry is recorded and status is displayed on an **LCD**. Data is sent to **Thingspeak** and **Blynk** for monitoring. • It reduces manual errors and improves attendance management with IoT integration. **Sensors used:** RFID RC522 **Actuator used:** LCD, Buzzr, LED **Cloud Platform:** Thingspeak, Blynk
17	**Title: Secure Access Control: RFID-based Smart Door Lock** **Synopsis:** • This project uses an **RFID RC522** sensor with **ATMEGA168/328** for door access. • Authorized users unlock the door via RFID, which triggers a **solenoid lock** and updates an **LCD** display. • Remote monitoring is available via **Blynk**, enhancing access security for homes and offices. **Sensors used:** RFID RC522 **Actuator used:** Solenoid Lock, LCD, Buzzer, LED **Cloud Platform:** Blynk, MQTT
18	**Title: Touch-Activated Controller: MQTT-based Device Interface** **Synopsis:** • **ATMEGA168/328** is integrated with a **touch sensor**, **buzzer**, and **relays** to control devices. • Touch input triggers the devices and sends real-time updates via **MQTT**, enabling low-latency IoT-based control. • Highlights efficient use of MQTT in responsive home automation systems. **Sensors used:** Touch Sensor (4 switches)

	Actuator used: Relay, Buzzer, LED **Cloud Platform:** MQTT, BlyNK **Database** :Firebase
19	**Title: Retro Metro Switch: Automated Switch Control** **Synopsis:** • This creative project repurposes a retro switch using two **servo motors** controlled by **ATMEGA168/328**. • The system mimics metro-style switching and allows remote control via **Blynk**, enhancing traditional infrastructure. • A smart approach to blending legacy tech with IoT automation. **Sensors used:** — **Actuator used:** Servo Motor (×2),LED, Buzzer **Cloud Platform:** Blynk, MQTT
20	**Title: Intelligent Entry: Smart Door Security and Access Control** **Synopsis:** • A **PIR sensor**, **servo motor**, and **buzzer** are connected to **ATMEGA168/328** for door control. • The PIR sensor detects motion and triggers servo-based door lock/unlock. • Alerts and monitoring are enabled via **Blynk**, enhancing home and office entry security. **Sensors used:** PIR Sensor **Actuator used:** Servo Motor, Buzzer, LED **Cloud Platform:** Blynk Database: Cloud Fire base
21	**Title: PillTime: Smart Medication Reminder and Tracker** **Synopsis:** • This project combines a **Real Time Clock (RTC)**, **LCD**, and **RF module** with **ATMEGA168/328** to remind users about their medication schedules. • It displays scheduled pill times on an LCD and sends reminders via **Blynk** for timely medication intake. • Especially beneficial in healthcare to improve adherence to prescribed routines. **Sensors used:** RTC **Actuator used:** LCD, RF Module, LED **Cloud Platform:** Blynk, MQTT
22	**Title: IoT Robotic Arm: Remote Manipulation and Control** **Synopsis:** • Using multiple **servo motors** interfaced with **ATMEGA168/328**, this project creates a remotely controllable robotic arm. • Movements are controlled via **Blynk**, enabling precise automation for education, industry, or remote handling.

	• Demonstrates effective integration of mechanics and IoT. **Sensors used:** — **Actuator used:** Servo Motors, Buzzer, LED **Cloud Platform:** Blynk, MQTT
23	**Title: Smart Scale: IoT-enabled Digital Weighing System** **Synopsis:** • This project uses a **load cell** with **ATMEGA168/328** to build a digital weighing scale. • Weight is displayed on an **LCD** and transmitted to **Blynk** for monitoring and analysis. • Ideal for commercial or personal use, combining accuracy and IoT analytics. **Sensors used:** Load Cell **Actuator used:** LCD, LED, Buzzer **Cloud Platform:** Blynk, MQTT
24	**Title: ColorSort Pro: Automated Color-based Sorting System** **Synopsis:** • A **color sensor** attached to **ATMEGA168/328** identifies object colors and sorts them accordingly. • An **OLED display** shows the detected color, while **MQTT** handles cloud-based monitoring. • Used in recycling or quality control for industrial automation. **Sensors used:** Color Sensor **Actuator used:** OLED,LED, Buzzer **Cloud Platform:** MQTT **Database:** Google Firebase
25	**Title: Smart Traffic Manager: IoT-Enabled 4-Way Signal Control** **Synopsis:** • This project uses an **IR sensor** and **LEDs** with **ATMEGA168/328** to manage traffic at a 4-way signal. • Sensors detect vehicle presence and optimize signal timing. • Integration with **Blynk** allows remote monitoring, improving traffic flow and safety. **Sensors used:** IR Sensor **Actuator used:** LEDs, Buzzer **Cloud Platform:** Blynk, MQTT **Database :** Google Firebase
26	**Title: Smart Assistance Stick: Multi-sensor Navigation Aid** **Synopsis:** • Designed for visually impaired individuals, this **ATMEGA168/328** project uses **ultrasonic**, **fire**, **water**, and **IR sensors**. • Real-time alerts are sent via **Blynk**, assisting safe navigation. • A powerful example of assistive technology using IoT.

	Sensors used: Ultrasonic, Fire, Water, IR **Actuator used:** Buzzer, LED **Cloud Platform:** Blynk, MQTT
27	**Title: Smart Waste Sorter: Automated Waste Segregation System** **Synopsis:** • **IR sensor** and **servo motor** with **ATMEGA168/328** automatically sort waste into separate bins. • Sensor data is transmitted over **MQTT** for remote tracking. • Promotes smart recycling by reducing manual effort. **Sensors used:** IR Sensor **Actuator used:** Servo Motor, LED, Buzzer **Cloud Platform:** MQTT, Thingspeak
28	**Title: Washroom Sentinel: Feedback and Monitoring System** **Synopsis:** • Built using **ATMEGA168/328**, this project collects user feedback via a **switch** and displays it on an **LCD**. • Data is logged and can be analyzed via **Blynk** or exported to Excel. • Helps maintain public hygiene by analyzing satisfaction levels. **Sensors used:** Switch **Actuator used:** LCD, Buzzer, LED **Cloud Platform:** Blynk **Database** Google Firebase
29	**Title: FireGuard Pro: Automated Fire and Temperature Alert System** **Synopsis:** • **Fire** and **temperature sensors** are connected to **ATMEGA168/328** to continuously monitor safety conditions. • When thresholds are exceeded, alerts are sent via **SMTP email** (HTML/text/attachments). • Excellent for proactive safety in industrial and residential spaces. **Sensors used:** Fire Sensor, Temperature Sensor **Actuator used:** Buzzer, LED **Cloud Platform:** SMTP (Email Notifications)
30	**Title: DoorWatch: Real-Time Door Status Monitoring with Telegram Alerts** **Synopsis:** • This project uses **ATMEGA168/328** with door sensors to monitor and send alerts via **Telegram**. • Notifies users instantly of door access or breaches, and logs the data for review. • Enhances home and office security using messaging platforms. **Sensors used:** Magnetic Door Sensor **Actuator used:** Buzzer, LED

	Cloud Platform: Telegram
31	**Title: MotionSense Hub: Comprehensive Inertial and Temperature Monitoring** **Synopsis:** • Combines **MPU-6050** (accelerometer, gyroscope) with **ATMEGA168/328** to monitor motion and temperature. • Data is displayed in **Blynk** and **Thingspeak** dashboards. • Ideal for asset tracking, movement analysis, and safety monitoring. **Sensors used:** MPU-6050 **Actuator used:** LED, Buzzer **Cloud Platform:** Blynk, Thingspeak
32	**Title: CloudBME: Real-time Environmental Monitoring with BME280** **Synopsis:** • **BME280** sensor with **ATMEGA168/328** monitors temperature, pressure, and humidity. • Data is sent to a real-time cloud database for access via mobile/web dashboard. • Triggers alerts on deviations, useful for both home and industry. **Sensors used:** BME280 **Actuator used:** LED, Buzzer **Cloud Platform:** Real-time Database
33	**Title: FarmSense Secure: Firebase-Integrated Sensor Dashboard** **Synopsis:** • **ATMEGA168/328** with **DS18B20 temperature sensor** and **soil moisture sensor** sends data to **Firebase**. • Secure web dashboard with Firebase authentication allows only authorized access. • Ideal for smart farming and remote environment monitoring. **Sensors used:** DS18B20, Soil Moisture Sensor **Actuator used:** Buzzer, LED **Cloud Platform:** Bynl, Thingspeak Database: Firebase
34	**Title:** Smart Energy Meter / Smart Grid System **Synopsis:** • This project utilizes **ATMEGA168/328** integrated with a **ZMPT101B voltage sensor** and an **ACS712 current sensor** to measure real-time voltage and current from the electrical supply. • The measured data is processed to calculate **power consumption**, which is displayed locally on an **OLED display**. • For remote monitoring and analytics, the data is sent to **Blynk**, **Thingspeak**, and stored in **Firebase** as a real-time database. • The system helps in monitoring energy usage, detecting overloads or abnormal consumption patterns, and promotes efficient energy management in homes or industries. • It also supports grid-level applications where power quality and usage data from multiple

	locations can be aggregated and analyzed. **Sensors used:** ZMPT101B (Voltage Sensor), ACS712 (Current Sensor) **Actuator used:** OLED Display, LED, Buzzer **Cloud Platform:** Blynk, Thingspeak Database: Firebase
35	**Title: Smart Health Monitor: Pulse Rate, Heartbeat & Oxygen Level Tracking System** **Synopsis:** • This project is based on **ATMEGA328** and integrates biomedical sensors to measure **pulse rate**, **heart rate**, and **SpO_2 (oxygen saturation)** levels. • The real-time health data is displayed on an **OLED screen**, and abnormal values trigger alerts via a **buzzer** and **LED indicator**. • For remote health tracking and data logging, the readings are uploaded to **Blynk**, **Thingspeak**, and **Firebase Real-time Database**. • This system is particularly useful for **home healthcare**, **elderly monitoring**, and **telemedicine**, allowing patients and doctors to monitor vitals remotely with real-time alerts. • It ensures timely awareness of critical health changes and supports emergency response. **Sensors used:** Pulse Sensor (or MAX30100/MAX30102 for heart rate & SpO_2) **Actuator used:** OLED Display, Buzzer, LED **Cloud Platform:** Blynk, Thingspeak Database: Firebase (Real-time Database)

Guidelines for execution of syllabus and maintain quality of mini project are as follows:

Teaching Approach:

1. Modules 1, 2, and 3 shall be taught in parallel to provide an integrated learning experience.
2. After covering the theoretical concepts of ATmega328 microcontroller architecture and its features, programming fundamentals should be introduced.
3. Embedded C programming should be taught in the context of microcontroller programming.
4. Initial programming exercises should focus on developing a strong understanding of Embedded C syntax and structure specific to microcontroller applications.
5. Subsequently, fundamental programming related to GPIO, Timers, and Interrupts must be covered along with their practical implementation on the ATmega328 microcontroller.
6. Module 4 shall focus on interfacing various sensors, actuators, and communication protocols with the ATmega328 microcontroller, covering their working principles and practical implementation on the development board.
7. Hands-on experiments must be conducted for each interfacing to ensure practical understanding and testing of sensors, actuators, and communication protocols.
8. After thorough understanding and practical exposure to Modules 1, 2, 3, and 4, students should proceed with project selection and execution based on the ATmega328 microcontroller.

Project Topic Selection and Approval:

1. A mini-project group shall consist of a minimum of THREE (03)
2. Project topic selection and approval should be done by a panel of two expert faculty members from the department.
3. Each group must maintain a logbook to record weekly work progress in terms of milestones. The guiding faculty should provide remarks/comments weekly. Both students and faculty must sign the logbook every week.

Project Report Format:

1. The mini-project report should include the following sections:
    - Abstract (maximum 1 page summarizing the complete report)
    - Introduction (justification for project selection, applications, and existing commercial products)
    - Implementation (proposed specifications, block diagram/circuit diagram, working principle/operation)

- Results & Discussion (photographs, video links, recorded results such as waveforms/observations/demonstrations)
- Conclusion and Learning Outcomes
- References
- Participation in Mini-project competition/Technical Paper Presentation (TPP), etc.

2. The report should not exceed 10 pages. Reports should be stapled to avoid the use of plastic.

Term Work & Practical/Oral Examination Evaluation:

1. Minimum six (06) experiments based on Modules 1, 2, 3, and 4 should be performed.
2. One mini project must be implemented.
3. It is not compulsory to select a project from the given sample list; students are encouraged to propose innovative ideas.
4. Students must deliver a presentation and demonstrate their mini project during the Practical and Oral Examination (25 Marks Evaluation).
5. Project reports must be checked for plagiarism using software such as Turnitin or any equivalent tool.
6. Evaluation should consider each student's individual contribution, understanding, and knowledge gained during the project execution. Marks should be awarded accordingly.

Term Work & Evaluation Scheme (Total 50 Marks):

Component	Marks
Performance in Experiments & Evaluation by Guide	15
Performance in mini project & Evaluation by Guide	10
Performance in mini project & Evaluation by Review Committee Evaluation	10
Logbook and Quality of Project Report	05
Research Paper	10