PONDICHERRY UNIVERSITY
(A CENTRAL UNIVERSITY)
SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

REGULATIONS, CURRICULUM & SYLLABUS
(For Affiliated Colleges)

B.Sc (Honors) DEGREE PROGRAMME

B.Sc. Computer Science
(Honors with Research)

B.Sc. Computer Science
(Honors)

(Under the National Education Policy 2020)
Effective from the Academic Year (2023 - 2024)

November 2023

2	Dr. K. Suresh Joseph	Professor	Pondicherry University
3	Dr. M. Sathya	Asst. Professor	

Table of Contents

1. PREAMBLE

The Bachelor of Science (B.Sc.) programme in Computer Science is a dynamic and comprehensive academic journey designed to equip students with a strong foundation in the principles and practices of computing. Rooted in the ever-evolving field of technology, this programme is crafted to cultivate a deep understanding of computer science theories, algorithms, and applications.

The curriculum encompasses a balanced blend of foundational courses and specialized electives on experiential learning, offering opportunities for internships, industry projects, and participation in coding competitions. Students will engage in practical applications of their knowledge, honing their skills through hands-on experiences that mirror the challenges and demands of the rapidly evolving technological landscape.

Recognizing the global nature of technology, the B.Sc. in Computer Science incorporates an international perspective. Students will explore global technology trends, multicultural influences, and ethical considerations, preparing them to contribute responsibly to the global digital community.

The B.Sc. in Computer Science at Pondicherry University is a transformative educational experience that empowers students to become adept problem solvers, innovators, and leaders in the field of computer science. By fostering a passion for continuous learning and providing a solid foundation in both theory and application, the programme sets the stage for a successful and fulfilling career in the dynamic world of technology.

2. PROGRAMME OUTCOMES

Upon completion of the Bachelor of Science (B.Sc.) programme in Computer Science, students will demonstrate the following outcomes at:

UG Certificate Level

- Acquire foundational knowledge in computer science.
- Demonstrate basic skills in problem-solving and programming.

UG Diploma Level

- Develop intermediate-level knowledge and skills in computer science.
- Apply problem-solving and programming concepts to practical scenarios.

UG Degree Level

- Attain advanced knowledge and skills in computer science.
- Demonstrate proficiency in problem-solving, programming, and system design.

UG Degree with Honors

- Demonstrate proficiency in programming languages and software development.
- Apply principles of data structures and algorithms to solve complex problems.
- Design and implement efficient solutions for real-world computing challenges.
- Exhibit effective communication skills in conveying technical concepts orally and in writing.
- Engage in collaborative projects and demonstrate the ability to work effectively in a team.
- Apply ethical considerations in professional and societal contexts related to computer science.
- Possess a comprehensive understanding on their Specialization in CCS / AI & ML / DS /
- C & DS / IT / CA.
- Exhibit a commitment to lifelong learning and adaptability to evolving technologies.

3. DEFINITIONS

Terms used in the NEP Regulations shall have the meaning assigned to them as given below unless the context otherwise requires:

A. Credit: A credit is the number of hours of instruction required per week for the given subject in a given semester of 16-18 weeks. One credit is equivalent to 15 hours of teaching (lecture or tutorial) or 30 hours of practice or field work or community engagement and service per Semester.

B. Academic Year: Means the year starting on 1st day of July and ends on the 30th day of June succeeding year.

C. Residence time: Means the time a student spends for attending classes in the College/Institution (either Online/Offline) as a full-time student and enrolled in any Academic programme of the Institution.

D. **Semester:** Means 18 weeks (90 Working days) of teaching-learning session of which two weeks shall be set apart for examinations and evaluation.

E. Grade: Means a letter grade assigned to a student in a course for his/her performance at academic sessions as denoted in symbols of: O(Outstanding), A+(Excellent), A(Very good), B+(Good), B(Above average), C(Average), P(Pass), F(Fail) and Ab(Absent) with a numeric value of O=10, A+=9, A=8, B+=7, B=6, C=5, P=4, and F=0, Ab=0.

F. Grade Point Average (GPA): Means an average of the Grades secured by a student in all courses in a given academic session duly weighted by the number of credits associated to each of the courses.

G. Cumulative GPA (CGPA): Means the weighted average of all courses the student has taken in the entire programme of study.

H. Common courses: Means the set of courses that all students who are admitted are required to study; these courses include, Languages (English- Modern Indian languages), NEP specific courses viz. Understanding India, Environmental sciences/Education, Health and wellbeing/Yoga, and Digital & Technological solutions.

I. Major Discipline Courses: Means the core subjects mandatory for the Computer Science discipline. These courses are common across all specializations of Computer Science.

J. Minor Discipline Courses: Means allied/elective/specialization specific subjects of Computer Science discipline. Based on the set of Minor Discipline Courses the candidate study, specialization in Computer Science will be awarded. Eg: B.Sc. (Computer Science) with minor discipline courses in Artificial Intelligence and Machine Learning will be awarded B.Sc. Computer Science with Specialization in AI&ML.

K. Credit Requirements: For a Degree/Diploma/Certificate Programme means the minimum number of credits that a student shall accumulate to achieve the status of being qualified to receive the said Degree, Diploma/Certificate as the case may be.

L. Exit option: Means the option exercised by the student, to leave the Programme at the end of any given Academic year.

M: Lateral entry: Means a student being admitted into an ongoing Programme of the University otherwise than in the 1st year of the programme.

N: Vocational Studies/Education: Means set of activities for participation in an approved project or practical or lab, practices of application of scientific theories, studio activities involving students in creative artistic activities, workshop-based activities, field-based shop-floor learning, and Community engagement services, etc. **(These courses are expected to enable students to incorporate the learned skills in daily life and start up entrepreneurship.)**

O: Skill-based learning/project: Means activities designed to understand the different socio-economic contexts, first-hand understanding of the policies, regulations, organizational structures, processes, and programmes that guide the development process.

P: Work-based internship: Means structured internships with Software Companies, Research and Higher Educational Institution Laboratories, Corporate offices, etc. which will further improve employability.

4. AWARD OF UG DEGREE/DIPLOMA/CERTIFICATE

Four years B.Sc. Degree Programme shall have options for earning a Certificate / Diploma / UG Degree / UG Degree with Honors based on the exit option exercised by the candidates.

4.1 Degree and Nomenclature

Candidates who complete Eight semesters and earn a minimum of 160 credits will be awarded either of the following degrees after successful completion of the other requirements.

- B.Sc. Computer Science (Honors with Research) *
- B.Sc. Computer Science (Honors) **

* for candidates who complete a research project work in the Eighth Semester

** for candidates who complete 3 theory courses (MJD 21, MJD 22, and MJD 23) instead of the research project work in the Eighth Semester

4.2 Exit Options

Candidates can exercise the following exit options and obtain the said certificate or diploma or degree, if the minimum required credits are earned and other conditions are met.

Exit after 2nd Semester: Certificate in Computer Science will be awarded for candidates who exit the course at the end of 2nd semester and earned a minimum of 40 credits and have completed a Summer Internship of 4 credits for 4 – 6 weeks duration, during the summer vacation post 2nd semester.

Exit after 4th Semester: Diploma in Computer Science will be awarded for candidates who exit the course at the end of 4th semester and earned a minimum of 80 credits and have completed a Summer Internship of 4 credits for 4 – 6 weeks duration, during the summer vacation post 4th semester.

Exit after 6th Semester: UG Degree in Computer Science (B.Sc. (CS)) will be awarded for candidates who exit the course at the end of 6th semester and earned a minimum of 120 credits and have completed a Summer Internship of 4 credits for 4 – 6 weeks duration, during the summer vacation post 4th semester.

Exit after	Credits and other requirements	Awards
2nd Semester	Min: 40 Credits, Internship 45 days	Certificate in Computer Science
4th Semester	Min: 80 Credits, Internship 45 days	Diploma in Computer Science
6th Semester	Min: 120 Credits, Internship 45 days	B.Sc. Computer Science

5. PEDAGOGICAL APPROACHES

COURSE TYPES	APPROACH
a) Lecture Courses	Regular classroom lectures by qualified / experienced Expert Teachers • These Lectures may also include classroom discussion, demonstrations, case analysis • Use of Models, Audio-Visual contents, Documentaries, PPTs may supplement.
b) Tutorial Courses	Problem solving Exercise classes guided discussion, supplementary readings vocational training, etc.
c) Practical / Lab work	Practical Lab activity with Theoretical support Mini projects, Activity based engagement, Program executions, Data processing and presentation exercise.
d) Seminar Course	A course requiring student to design and participate in discussions, Group Discussions, Elocution and Debate, Oral Communication Paper presentations, Poster Presentation, Role play participation, Quiz competitions, Business plan preparation/presentation, etc.

e) Internship course	Courses requiring students to *Learn by Doing* in the workplace external to the educational Institutions. Internships involve working in Software Companies, Research and Higher Educational Institution Laboratories, Corporate Offices, etc. All Internships should be properly guided and inducted for focused learning.
f) Research Project	Students need to study and analyze the recent research publications from indexed/peer reviewed journals in their area of specialization. Outcome of the study and analysis need to be presented as a thesis or research report with necessary experimental results.

6. ACADEMIC AUDIT OF COURSES

Internal Quality Assurance Cell (IQAC) at every institution is expected to supervise the implementation of NEP Regulations in these programmes. Availability of required number of Classrooms, Faculty rooms, Labs, Library facilities, Computer Centre and recruitment of Faculty members, allocation of funds for running the Science Labs/Computer Centre etc., is the responsibility of University / College Administration.

7. ADMISSIONS & LATERAL ENTRY

7.1 Admissions Eligibility

The candidates for admission to this programme shall be required to have passed 10+2 / 10+3 system of examinations or equivalent with mathematics / business mathematics / equivalent as one of the subjects of study.

Students shall be admitted to this programme based on admissions criteria fixed by the University / Government of Puducherry from time to time.

7.2 Lateral Entry

As per NEP, students have a choice of exit and entry into the programme multiple number of times. UGC specifies that about 10% of seats over and above the sanctioned strength shall be allocated to accommodate the Lateral Entry students.

Candidates seeking entry at the second, third and fourth year, should meet the necessary eligibility criteria with respect to the certificate / diploma / degree they possess, with necessary minimum credits banked in the Academic Bank of Credits (ABC). Such students who get admitted in later years, other than first year will be guided by the following clauses:

- that the University shall notify the admission process and number of vacancies open for lateral entry.
- that the Lateral entrants shall be admitted only after such transparent screening process and such procedure that the University may prescribe from time to time. University may

prescribe different methods of screening for different programmes depending on the circumstances prevailing in each case.

- Lateral entry shall be permissible only in the beginning of years 2, 3, 4 of the Under Graduate / Honors programme; provided that the students seeking lateral entry shall have obtained the minimum pass marks / grades fixed by the University in their previous academic years.

8. EVALUATION (INTERNAL & END SEMESTER ASSESSMENT) AND GRADES

All Credit courses are evaluated for 100 marks. Internal Assessment component is for 25 marks and the End Semester University exam is for 75 marks for theory courses. In case of practical courses, research project work etc., Internal Assessment component is for 50 marks and the End Semester University exam is for 50 marks.

Internal Test Scheme: Principal of the College schedules the Mid-Semester Exam for all courses during 8/9th week of start of classes. Mid-Semester exam for 90 minutes' duration need to be conducted for all these theory courses. The evaluated marks need to be uploaded to Controller of Examinations of University. The answer books of Mid-Semester exams need to be preserved until the declaration of results by the University.

8.1 INTERNAL ASSESSMENTS (for Courses upto 6th Semester)

8.1.1 Internal Assessment Marks for Theory subjects

Total Internal Assessment mark for a theory subject is 25 marks. The breakup is as follows:

Evaluation Component	Marks
A. Mid Semester Exam (one)	20
B. Percentage of Attendance	05
Total	**25**

8.1.2 Internal Assessment marks for Practical / Internships subjects

Faculty member in-charge of Lab practical shall evaluate the practical subjects for 50 marks. The breakup is as follows:

Evaluation Component	Marks
A. Mid-Semester Practical Exam (one) / Viva-voce	20

B. Practical Record / Internship Report	25
C. Percentage of Attendance	05
Total	**50**

8.1.3 Internal Assessment marks for Research Project Work

There shall be a faculty member assigned as a Project Guide for each candidate doing the Research Project. Progress of the candidate can be assessed once in a month in a project review meeting. Three project review meetings shall be conducted for Internal Assessment.

Project review committee may be constituted and the committee shall organize project review meetings and evaluate the progress and to award the Internal Assessment marks. Internal Assessment component for the Research Project is 50 Marks. The breakup is as follows:

Evaluation Component	Marks
A. Monthly Review (3 Reviews – 10 Marks each)	30
B. Project Report	10
C. Project Presentation and viva-voce	10
Total	**50**

8.1.4 Internal Assessment marks for Theory Subjects with Practical Components

Faculty member in-charge of Theory Subjects with Practical Component shall evaluate the candidates both for their performance in theory and practical. Internal Assessment marks for Theory Subjects with Practical Components is 25 marks. The breakup is as follows:

Evaluation Component	Marks
A. Mid Semester Exam (one)	15
B. Observation Note / Practical Record	05
C. Percentage of Attendance	05
Total	**25**

8.1.5 Marks for Attendance is as follows

Attendance %	Marks
Below 75%	0
75% - 80%	1
80% - 85%	2
85% - 90%	3
90% - 95%	4
95% - 100%	5

8.2 END SEMESTER ASSESSMENT (for Courses upto 6th Semester)

Controller of Examinations (COE) of Pondicherry University schedules the End-Semester exams for all theory and practical subjects based on university calendar. For Theory courses with Practical components, End semester exams shall be conducted separately for Theory and Practical.

A detailed Exam Time Table shall be circulated at least 15 days before the start of exams, mostly during 15/16th week of the Semester. Question Papers shall be set externally based on BoS approved syllabus. All students who have a minimum of 70% attendance are eligible to attend the end-semester exams. Attendance percentage shall be calculated for each course to decide the eligibility of the candidate for writing the end-semester examination.

8.2.1 Breakup of End Semester Marks
(All End Semester Exams shall be conducted by the Pondicherry University)

The question paper shall be set as per the Bloom's Taxonomy. Various levels along with it's description and sample questions are as follows:

Knowledge: Recall or remember previously learned information.
Example: List the basic data types in Python

Comprehension: Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions, and stating the main ideas.
Example: Explain how a stack data structure works.

Application: Apply knowledge and concepts to solve problems in new situations. Use learned information in a different context.
Example: Write a Python program to solve the deadlock problem.

Analysis: Break down information into parts and examine the relationships between the parts. Identify motives or causes.
Example: Analyse the efficiency of two sorting algorithms and compare their advantages and disadvantages.

Synthesis: Create a new whole by combining elements in novel ways. Use creativity to produce something original.
Example: Design a web application that can generate a time table of a school.

Distribution of questions at various levels are as indicated.

Course Components	Max. Marks	End-Sem Exam Duration
A. Theory subjects: Sec A: 10 Questions of 2 Marks each (20 Marks) *(Knowledge: 3, Comprehension: 2, Application: 3, Analysis:2)* Sec B: 5 out of 7 Questions of 5 Marks each (25 Marks) *(Knowledge: 1, Comprehension: 2, Application: 1, Analysis:3)* Sec C: 2 Either/OR choice questions of 15 Marks each (30 Marks) *(Application: 1, Analysis:1)* **Questions from all units of Syllabus equally distributed.**	75 Marks	3 Hours
B. Skill Enhancement/ Practical/Internship/Project Work subjects: **Skill Enhancement / Practical Subjects:** Based on Practical Exams conducted by CoE of University **Internship / Research Project Work:** Presentation of the work / Report / Viva-voce examinations	50 Marks	3 Hours --
C. Theory Subjects with Practical Components: **i. Theory Component:** Sec A: 5 Questions of 2 Marks each (10 Marks) *(Knowledge: 3, Comprehension: 2, Application: 3, Analysis:2)* Sec B: 5 out of 7 Questions of 4 Marks each (20 Marks) *(Comprehension: 2, Application: 3, Analysis:2)* Sec C: 2 Either or type questions of 10 Marks each (20 Marks) *(Analysis / Synthesis)* **Questions from all units of Syllabus equally distributed.** **ii. Practical Component:** Based on Practical Exams / Presentation / Viva-voce with external examiner appointed by the University Controller of Examinations, and schedules exclusively prepared for such practical examinations by the University Examination Section. **The examination shall be conducted for 50 Marks and reduced to 25 Marks.** **Total Marks: 75 (Theory: 50 Marks + Practical: 25 Marks)**	50 Marks 25 Marks	3 Hours 3 Hours

8.3 CONSOLIDATION OF MARKS AND PASSING MINIMUM

Controller of Examinations of the University consolidates the Internal Assessment marks uploaded by the Colleges and marks secured by students in End-Semester examinations. The total marks will be converted into letter grades. The passing minimum is 40% marks (Internal Assessment + End Semester Assessment put together) and students who secure between 40% and 49% will be awarded 'P' (Pass Grade).

8.3.1 Arrear Exam

A student who secures less than 40% marks in aggregate is declared as ***Fail*** and that student is eligible to take up supplementary examination by registering to the failed course in the following Semester. All other candidates who failed due to shortage of attendance and those who are seeking to improve the grade shall repeat the course.

8.3.2 Letter Grades and Calculation of CGPA

Total marks secured by a student in each subject shall be converted into a letter grade. UGC Framework has suggested a Country wide uniform letter grades for all UG courses. The following table shows the seven letter grades and corresponding meaning and the grade points for calculation of CGPA.

Equivalent Letter Grade	Meaning	Grade Points for Calculation of CGPA
O	Outstanding	10
A+	Excellent	9
A	Very Good	8
B+	Good	7
B	Above Average	6
C	Average	5
P	Pass	4
F	Fail	0
Ab	Absent	0

In order to work out the above letter grades, the marks secured by a student (Total of Internal Assessment and End Semester Assessment) would be categorized for relative grading.

The range of marks for each grade would be worked as follows:

- Highest marks in the given subject: X

- Cut of marks for grading purpose: 50 marks
- Passing minimum: 40
- Number of grades (except P - Pass) (O, A+, A, B+, B, C): G = 6
- Range of marks: K = (X - 50) / G

(i) If K ≥ 5, then the grades shall be awarded as given in the following table.

Range of Marks in %	**Letter Grade Points for**	**Grade Points for**
X to (X-K) + 1	O	10
(X-K) to (X-2K) + 1	A+	9
(X-2K) to (X-3K) + 1	A	8
(X-3K) to (X-4K) + 1	B+	7
(X-4K) to (X-5K) + 1	B	6
(X-5K) to 50	C	5
40 – 49	P	4
Below 40	F	0
Absent (Lack of Attendance)	Ab	0

(ii) If K< 5, then the grades shall be awarded as given in the following table.

Range of Marks in %	**Letter Grade Points for**	**Grade Points for**
80-100	O	10
71-79	A+	9
66-70	A	8
61-65	B+	7
56-60	B	6
50-55	C	5
40-49	P	4
Below 40	F	0
Absent (lack of attendance)	Ab	0

8.3.3 Calculation of Semester Grade Point Average and Cumulative Grade Point Average

Semester Grade Point Average (SGPA) is calculated by taking a weighted average of all grade points secured by a candidate from all subjects registered by him/her in the given Semester. The weights being the number of credits that each subject carries.

Cumulative Grade Point Average (CGPA) shall be calculated as the weighted average of credits that course carries and the value of Grade points averaged for all subjects.

8.3.4 Computation of SGPA and CGPA

The following procedure shall be followed to compute the Semester Grade Point Average (SGPA) and Cumulative Grade Point Average (CGPA):
The SGPA is the ratio of the sum of the product of the number of credits with the grade points scored by a student in all the courses taken by a student to the sum of the number of credits of all the courses undergone by a student, i.e. SGPA $(S_i) = \Sigma(C_i \times G_i) / \Sigma C_i$

where C_i is the number of credits of the i^{th} course and G_i is the grade point scored by the student in the i^{th} course.

(i) Example for Computation of SGPA where candidate has not failed in any course

Semester	**Course**	**Credit**	**Letter Grade**	**Grade point**	**Credit Point** (Credit x Grade)
I	Course 1	3	A	8	3 X 8 = 24
I	Course 2	4	B+	7	4 X 7 = 28
I	Course 3	3	B	6	3 X 6 = 18
I	Course 4	3	O	10	3 X 10 = 30
I	Course 5	3	C	5	3 X 5 = 15
I	Course 6	4	B	6	4 X 6 = 24
		20			139
				SGPA	**139/20=6.95**

(ii) Example for Computation of SGPA where candidate has failed in one course

Semester	**Course**	**Credit**	**Letter Grade**	**Grade point**	**Credit Point** (Credit x Grade)
I	Course 1	3	A	8	3 X 8 = 24
I	Course 2	4	B+	7	4 X 7 = 28
I	Course 3	3	B	6	3 X 6 = 18
I	Course 4	3	O	10	3 X 10 = 30
I	Course 5	3	C	5	3 X 5 = 15
I	Course 6	4	F	0	4 X 0 = 00
		20			115
				SGPA	**115/20=5.75**

(iii) Example for Computation of SGPA where candidate has failed in two courses

Semester	Course	Credit	Letter Grade	Grade point	Credit Point (Credit x Grade)
I	Course 1	3	A	8	3 X 8 = 24
I	Course 2	4	B+	7	4 X 7 = 28
I	Course 3	3	F	0	3 X 0 = 00
I	Course 4	3	B	6	3 X 6 = 18
I	Course 5	3	C	5	3 X 5 = 15
I	Course 6	4	F	0	4 X 0 = 00
		20			85
				SGPA	**85/20=4.25**

The CGPA shall also be calculated in similar way as shown in examples (i), (ii) and (iii) of SGPA for all subjects taken by the students in all the semesters. However, if any student fails more than once in the same subject, then while calculating CGPA, the credit and grade point related to the subject in which the student fails in multiple attempts will be restricted to one time only. The SGPA and CGPA shall be rounded off to 2 decimal points and reported in the transcripts.

In case of audit courses offered, the students may be given (P) or (F) grade without any credits. This may be indicated in the mark sheet. Audit courses will not be considered towards the calculation of CGPA.

8.3.5 Declaration of Results

Controller of Examinations (COE) of the University shall declare the results of given UG programme following the CGPA secured by students by the end of 6th Semester and 8th Semester.

8.3.6 Classification of Divisions

Range of CGPA	Result
9.0 – 10	First Class with distinction$^{\#}$
6.0 - 8.99	First Class
5.0 - 5.99	Second Class
4.0 - 4.99	Pass

Distinction will be awarded ONLY to those candidates who have cleared ALL subjects in the first attempt.

8.4 INTERNAL ASSESSMENT / END-SEMESTER ASSESSMENT / PASSING MINIMUM / GRADES (FOR 7TH & 8TH SEMESTERS)

Regulations to be notified in the next revision after the confirmation from University NEP committee.

9. MINIMUM CREDIT REQUIREMENTS

S.No	Component	3-year UG			4-year UG (Honors / Honors With research)		
		Credits	Courses	Cr/Course	Credits	Courses	Cr/Course
1	Major Disciplinary/ Interdisciplinary Courses	56	14	4	76	19	4
2	Minor Disciplinary/ Interdisciplinary Courses	24	6	4	32	8	4
3	Multi-Disciplinary Courses	9	3	3	9	3	3
4	Ability Enhancement Courses	8	4	2	8	4	2
5	Skill Enhancement Courses	9	3	3	9	3	3
6	Value-added courses	8	4	2	8	4	2
7	Summer Internship (MJD 11)	4	1	4	4	1	4
8	Community Engagement and Service	2	1	2	2	1	2
9	Research Project/Dissertation	--	--	--	12	Project or 3 Courses##	
Total		**120**			**160**		

##Note: *Honors students not undertaking research will do 3 courses for 12credits in lieu of a research project/Dissertation.*

- *MJD: Major Disciplinary (Compulsory – Hardcore Subjects)*
- *MID: Minor Disciplinary (Specialization Specific – Softcore Subjects)*
- *MLD: Multi-Disciplinary*
- *AEC: Ability Enhancement Courses*
- *SEC: Skill Enhancement Courses*
- *VAC: Value Added Courses*
- *Course Code:* CS1MJ01(E) *(BCS-B.Sc Computer Science, 1-Semester, MJ-Component, 1-Course Number in the respective component, E-Elective)*

B.Sc. COMPUTER SCIENCE

CURRICULUM

FIRST SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 1	CS1MJ01	Digital Logic Fundamentals	H	4	3		2
2	MID 1	CS1MI01	Microprocessor and ALP	S	4	3		2
3	MLD 1		One course from the MLD streams 1 to 10 (Table 15)	H	3	4		
4	AEC 1	CS1AE01	English I	H	2	2		2
5	SEC 1		S.No. 1 or 2 from Table 7	S	3	2		2
6	VAC 1	CS1VA01	Understanding India	H	2	4		0
7	VAC 2	CS1VA02	Environmental Sciences/Education	H	2	4		0
				Total	20	30 Hours		

SECOND SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 2	CS2MJ02	Problem Solving & Programming Fundamentals	H	4	3		2
2	MID 2	CS2MI02	Microcontrollers Programming	S	4	3		2
3	MLD 2		One course from the MLD streams 1 to 10 except the stream chosen in MLD1 (Table 15)	H	3	4		
4	AEC 2	CS2AE02	Indian Language I	H	2	2		2
5	SEC 2		S.No. 3 or 4 from Table 7	S	3	2		2
6	VAC 3	CS2VA03	Health & Wellness/Yoga Education	H	2			4
7	VAC 4	CS2VA04	Digital Technologies	H	2	3		
				Total	20	29 Hours		

THIRD SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 3	CS3MJ03	Mathematical Foundations of CS	H	4	4	1	
2	MJD 4	CS3MJ04	Data Structures	H	4	3		2
3	MID 3	CS3MI03	System Software	S	4	3		2
4	MLD 3		One course from the MLD streams 1 to 10 except the streams chosen in MLD1 and MLD2 (Table 15)	H	3	4		
5	AEC 3	CS3AE03	English II	H	2	2		2
6	SEC 3		S.No. 5 or 6 from Table 7	S	3	2		2

Total	**20**	**27 Hours**

FOURTH SEMESTER								
S.No.	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MJD 5	CS4MJ05	Computer System Architecture	H	4	3		2
2	MJD 6	CS4MJ06	Design and Analysis of Algorithms	H	4	3		2
3	MJD 7	CS4MJ07	Object Oriented Programming	H	4	3		2
4	MID 4	CS4MI04	Embedded Application Development	S	4	3		2
5	AEC 4	CS4AE04	Indian Language II	H	2	2		2
6	Project	CS4CS01	Community Engagement and Service	H	2			6
Total					**20**	**30 Hours**		

FIFTH SEMESTER								
S.No.	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MJD 8	CS5MJ08	Operating Systems	H	4	3		2
2	MJD 9	CS5MJ09	Database Management Systems	H	4	3		2
3	MJD 10	CS5MJ10	Management Strategies & Concepts	H	4	4		
4	MID 5	CS5MI05	Theory of Computation	S	4	3	2	
5	MJD 11	CS5MJ11	Summer Internship	H	4			6
Total					20	25 Hours		

SIXTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MJD 12	CS6MJ12	Computer Networks	H	4	3		2
2	MJD 13	CS6MJ13	Software Engineering	H	4	3		2
3	MJD 14	CS6MJ14	System Modelling & Simulation	H	4	3		2
4	MJD 15	CS6MJ15	Web Engineering	H	4	3	2	
5	MID 6		Any one course from Table 1	S	4	3		2
Total					20	25 Hours		

SEVENTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MJD 16	CS7MJ16	Software Testing and Quality Assurance	H	4	3		2
2	MJD 17	CS7MJ17	Distributed Systems	H	4	3		2
3	MJD 18	CS7MJ18	Wireless Communication Networks (5G)	H	4	3		2
4	MID 7		Any one course from Table 2	S	4	3		2
5	MID 8		Any one course from Table 3	S	4	3		2

Total	**20**	**25 Hours**

EIGHTH SEMESTER – B.Sc. Computer Science (Honors)								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MJD 19		Any one course from Table 4	S	4	3		2
2	MJD 20		Any one course from Table 5	S	4	3		2
3	MJD 21	CS8MJ21	High Performance Computing	H	4	3		2
4	MJD 22	CS8MJ22	Cloud Computing	H	4	3		2
5	MJD 23	CS8MJ23	Deep Learning	H	4	3		2
			Total		20	25 Hours		

EIGHTH SEMESTER – B.Sc. Computer Science (Honors with Research)								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MJD 19		Any one course from Table 4	S	4	3		2
2	MJD 20		Any one course from Table 5	S	4	3		2
3	MJD 21	CS8MJ24	Research Project	H	4			5
4	MJD 22	CS8MJ25	Project Report	H	4			5
5	MJD 23	CS8MJ26	Project Viva-voce	H	4			5
			Total		20	25 Hours		

Table 1: MID 6 – SIXTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MID 6	CS7MI06E1	Unix System Programming	S	4	3		2
2	MID 6	CS7MI06E2	Network Programming	S	4	3		2

Table 2: MID 7 – SEVENTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MID 7	CS7MI07E1	Artificial Intelligence	S	4	3		2
2	MID 7	CS7MI07E2	Compiler Design	S	4	3		2

Table 3: MID 8 – SEVENTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	**MID 8**	**CS7MI08E1**	**Cyber Security**	**S**	**4**	**3**		**2**
2	**MID 8**	**CS7MI08E2**	**Internet of Things**	**S**	**4**	**3**		**2**

Table 4: MJD 19 – EIGHTH SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 19	CS8MJ19E1	Machine Learning	S	4	3		2
2	MJD 19	CS8MJ19E2	Full Stack Development	S	4	3		2
3	MJD 19	Cs8MJ19E3	ADBMS					

Table 5: MJD 20 – EIGHTH SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 20	CS8MJ20E1	Wireless Sensor Networks	S	4	3		2
2	MJD 20	CS8MJ20E2	IOT	S	4	3		2
2	MJD 20	CS8MJ20E3	Data Mining	S	4	3		2

Table 6: MJD 21 / MJD 22 / MJD 23 – EIGHTH SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 21	CS8MJ21	High Performance Computing	H	4	3		2
2	MJD 22	CS8MJ22	Cloud Computing	H	4	3		2
3	MJD 23	CS8MJ23	Deep Learning	H	4	3		2

Table 7: SEC 1 / SEC 2 / SEC 3 – I / II / III SEMESTERs								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	SEC 1	CS1SE01E1	Python Programming	S	3	3		2
2	SEC 1	CS1SE01E2	R Programming	S	3	3		2
3	SEC 2	CS2SE02E1	Programming for Mobile Devices	S	3	3		2
4	SEC 2	CS2SE02E2	Visual Programming with C#	S	3	3		2
5	SEC 3	CS3SE03E1	Game Programming	S	3	3		2
6	SEC 3	CS3SE03E2	3D Modelling & Animation	S	3	3		2

Table 8: List of Major Disciplinary Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**MJD 1**	**CS1MJ01**	Digital Logic Fundamentals	**H**
2.	**MJD 2**	**CS2MJ02**	Problem Solving & Programming Fundamentals	**H**
3.	**MJD 3**	**CS3MJ03**	Mathematical Foundations of Computer Science	**H**
4.	**MJD 4**	**CS3MJ04**	Data Structures	**H**
5.	**MJD 5**	**CS4MJ05**	Computer System Architecture	**H**
6.	**MJD 6**	**CS4MJ06**	Design and Analysis of Algorithms	**H**
7.	**MJD 7**	**CS4MJ07**	Object Oriented Programming	**H**
8.	**MJD 8**	**CS5MJ08**	Operating Systems	**H**
9.	**MJD 9**	**CS5MJ09**	Database Management Systems	**H**
10.	**MJD 10**	**CS5MJ10**	Management Strategies & Concepts	**H**
11.	**MJD 11**	**CS5MJ11**	Summer Internship	**H**
12.	**MJD 12**	**CS6MJ12**	Computer Networks	**H**
13.	**MJD 13**	**CS6MJ13**	Software Engineering	**H**
14.	**MJD 14**	**CS6MJ14**	System Modeling & Simulation	**H**
15.	**MJD 15**	**CS6MJ15**	Web Engineering	**H**
16.	**MJD 16**	**CS7MJ16**	Software Testing and Quality Assurance	**H**
17.	**MJD 17**	**CS7MJ17**	Distributed Systems	**H**
18.	**MJD 18**	**CS7MJ18**	Wireless Communication Networks (5G)	**H**
19.	**MJD 19**		Machine Learning / Full Stack Development / Advanced Database Management Systems	**S**
20.	**MJD 20**		WSN / IoT / Data Mining	**S**

Table 9: List of Minor Disciplinary Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**MID 1**	**CS1MI01**	Microprocessor and ALP	**S**
2.	**MID 2**	**CS2MI02**	Microcontrollers Programming	**S**
3.	**MID 3**	**CS3MI03**	System Software	**S**
4.	**MID 4**	**CS4MI04**	Embedded Application Development	**S**
5.	**MID 5**	**CS5MI05**	Theory of Computation	**S**
6.	**MID 6**	**CS6MI06**	UNIX System Programming / Network Programming	**S**
7.	**MID 7**		AI / Compiler Design	**S**
8.	**MID 8**		Cyber Security / Internet of Things	**S**

Table 10: List of Multi-disciplinary Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**MLD 1**	**CS1ML01**	Natural Sciences	**H**
2.	**MLD 2**	**CS2ML02**	Physical Sciences	**H**
3.	**MLD 3**	**CS3ML03**	Humanities & Social Sciences	**H**

Table 11: List of Ability Enhancement Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**AEC 1**	**CS1AE01**	English I	**H**
2.	**AEC 2**	**CS2AE02**	Indian Language I	**H**
3.	**AEC 3**	**CS3AE03**	English II	**H**
3.	**AEC 4**	**CS4AE04**	Indian Language II	**H**

Table 12: List of Skill Enhancement Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**SEC 1**	**CS1SE01E1**	Introduction to Python Programming	**S**
2.	**SEC 1**	**CS1SE01E2**	R Programming	**S**
3.	**SEC 2**	**CS2SE02E1**	Programming for Mobile Devices	**S**
4.	**SEC 2**	**CS2SE02E2**	Visual Programming with C#	**S**
5.	**SEC 3**	**CS3SE03E1**	Game Programming	**S**
6.	**SEC 3**	**CS3SE03E2**	3D modeling and Animation	**S**

Table 13: List of Value-Added Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**VAC 1**	**CS1VA01**	Understanding India	**H**
2.	**VAC 2**	**CS1VA02**	Environmental Education	**H**
3.	**VAC 3**	**CS2VA03**	Health & Wellness / Yoga Education	**H**
4.	**VAC 4**	**CS2VA04**	Digital Technologies	**H**

Table 14: Project (WP/ Internship)				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**Project**	**CS4CS01**	Community Engagement and Service	**H**

*Table 15: MLD 1 / MLD 2 / MLD 3 in Sem 1 / Sem 2 / Sem 3				
S.No	**Streams**	**Course Code**	**Title of the Course**	**H/S**
1.	**Natural Science**		Biology	**H**
2.			Botany	**H**
3.			Zoology	**H**
4.			Biotechnology	**H**
5.			Biochemistry	**H**
6.	**Physical Sciences**		Chemistry	**H**
7.			Physics	**H**
8.			Biophysics	**H**
9.			Astronomy	**H**
10.			Astrophysics	**H**
11.			Earth and Environmental Sciences	**H**
12.	**Social Sciences**		Political Sciences	**H**
13.			History	**H**
14.			Social work	**H**
15.			Sociology	**H**
16.	**Humanities**		Anthropology	**H**
17.			Psychology	**H**
18.			Economics	**H**
19.	**Computer Science & Applications**	**CS1SE01E1 (ODD)**	Introduction to Python Programming	**H**
20.		**CS2SE02E1 (EVEN)**	Programming for Mobile Devices	**H**
21.				**H**

*Courses will be announced after the approval of the respective boards.

B.Sc. Computer Science

SYLLABUS

SEMESTER I

Year	I	Course Code: CS1MJ01	Credits	4
Sem.	I	Course Title: Digital Logic Fundamentals	Hours	75
Course Prerequisites, if any	Nil			
Internal Assessment Marks: 25	End Semester Marks: 75	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the postulates of Boolean algebra. • Apply minimization techniques for combinational functions. • Design and analyze combinational and sequential circuits. • Analyze and apply techniques for the design of digital circuits. • Create simple digital circuit designs and schematics.			
Unit No.	**Course Content**		**Hours**	
	Theory Component			
Unit I	**Digital Systems and Binary Numbers** Digital Systems - Binary Numbers - Number-Base Conversions - Octal and Hexadecimal Numbers - Complements of Numbers - Signed Binary Numbers - Binary Codes - Binary Storage and Registers - Binary Logic - Axiomatic Definition of Boolean Algebra - Basic Theorems and Properties of Boolean Algebra - Boolean Functions Canonical and Standard Forms - Other Logic Operations - Digital Logic Gates - Integrated Circuits		9	
Unit II	**Gate-Level Minimization** Introduction - The Map Method - Four-Variable K-Map - Product-of-Sums Simplification - Don't-Care Conditions - NAND and NOR Implementation - Other Two-Level Implementations - Exclusive-OR Function - Hardware Description Language		9	
Unit III	**Combinational Logic** Introduction - Combinational Circuits - Analysis Procedure - Design Procedure - Binary Adder–Subtractor - Decimal Adder - Binary Multiplier - Magnitude Comparator – Decoders – Encoders – Multiplexers - HDL Models of Combinational Circuits.		9	
Unit IV	**Synchronous Sequential Logic** Introduction - Sequential Circuits - Storage Elements: Latches - Storage Elements: Flip-Flops - Analysis of Clocked Sequential Circuits - Synthesizable HDL Models of Sequential Circuits - State Reduction and Assignment - Design Procedure		9	
Unit V	**Registers and Counters**		9	

	Registers - Shift Registers - Ripple Counters - Synchronous Counters - Other Counters - HDL for Registers and Counters	
Practical Component		
Exercises	1. Binary to Decimal and vice-versa in Python 2. Decimal to Hexadecimal and Vice-Versa in Python 3. Digital Logic Gates in Python 4. Simplification of Boolean Functions in Python 5. Combinational Logic Circuits in Python i. Code Converters ii. Arithmetic (Adders, Subtractors, Multipliers, Comparators) iii. Data Handling (Multiplexers, Demultiplexers, Encoders & Decoders) 6. Combinational Logic Circuit Design in Python 7. Binary Adder-Subtractor Simulation in Python 8. Decimal Adder Simulation in Python 9. Binary Multiplier Simulation in Python 10. Sequential Circuit Storage Elements: Flip-Flop Simulation in Python (Many more programs can be included related to programming the Digital logic in Python)	30
Recommended Learning Resources		
Print Resources	1. M. Morris Mano , Michael D. Ciletti, Digital design With an Introduction to the Verilog HDL, Pearson, Fifth Edition, 2013, ISBN-13: 978-0-13-277420-8, ISBN-10: 0-13-277420-8. 2. M. Rafiquzzaman, Fundamentals of Digital Logic and Microcomputer Design, John Wiley & Sons, Inc., Fifth Edition, 2005.	

Year	I	Course Code: CS1MI01	Credits	4
Sem.	I	Course Title: Microprocessor & ALP	Hours	75
Course Prerequisites, if any	Number Systems (binary, octal, hexadecimal) and their conversions. Boolean Algebra, logic gates, flip-flops and registers. Concepts in Combinational and Sequential logic.			
Internal Assessment Marks: 25	End Semester Marks: 75	Duration of ESA (Theory) : 03 hrs. Duration of ESA (Practical) : 03 hrs.		
Course Outcomes	• Learn the architecture & organization of 8085 Microprocessor. • Understand and classify the instruction set of the 8085 Microprocessor. • Apply the memory & I/O Interfacing with 8085 Microprocessor. • Analyse the architecture and operation of Programmable Interface. • Create applications to interface various peripheral IC's with Intel 8085 microprocessor.			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction to Microprocessors & 8085 Assembly Language Programming** Microprocessors, Instruction set and computer languages, 8085 programming model, Instruction classification, Instruction, Data format and storage, Execute a simple program , 8085 instruction set		9	
Unit II	**8085 Microprocessor architecture** Microprocessor Architecture and its operations, Memory, I/O Devices, 8085 MPU, 8085 based microcomputer, memory interfacing, 8155 memory segment Interfacing, Interfacing I/O devices: Basics, Interfacing input and output devices, memory mapped I/O		9	
Unit III	**Programming 8085** Instruction Set of 8085, Data Transfer, arithmetic, Logic, Branch, Writing ALP and Debugging programs, Looping, Counting and Indexing, 16 bit Arithmetic instructions, Logic operations, Counters and Time Delay		9	
Unit IV	**Interfacing I/O Devices** Stack and subroutines, Restart, Conditional call and Return instruction, Advanced subroutine concepts, Code conversion, BCD Arithmetic and 16 bit operations, BCD- Binary conversion, Binary to BCD conversion, BCD to seven segment LED code conversion, Binary to ASCII and ASCII to binary conversion, BCD addition and subtraction.		9	
Unit V	**Interfacing Peripheral (I/O) and Applications** Interrupts: 8085 Interrupt, RST instructions, Software and Hardware interrupt, multiple Interrupts and Priorities, 8085 Vectored Interrupts, Restart as Software Instructions, 8155 – Multipurpose programmable Device, 8279 – Programmable Keyboard/Display Interface, 8255 – Programmable peripheral Interface		9	
Practical Component				
Exercises	1. Assembly Language Programming for Arithmetic Operations like Addition, Subtraction, Multiplication and Division on 8, 16-bit data. 2. Assembly Language Programming for different logical operations. 3. Assembly Language Programming for code conversions. 4. Assembly Language Programming for sorting 5. Assembly Language Programming for Searching. 6. Assembly Language Programming for memory block transfer.		30	

	7. Assembly Language Programming using subroutines. 8. Assembly Language Programming using counters and time delay. (Many more programs can be included related to the programming techniques of Microprocessor 8085)	
Recommended Learning Resources		
Print Resources	1. Ramesh S. Gaonkar, Microprocessor – Architecture, Programming and Applications with the 8085, Penram International Publisher, 6th Edition 2013. 2. Douglas V. Hall, Microprocessors and Interfacing, Tata McGraw Hill publications, 2nd Edition, 2012. 3. Intel Corp: The 8085 / 8085A. Microprocessor Book – Intel marketing communications, Wiley Inter Science publications, 1980, (Digitized:17 Nov 2007) ISBN:0471035688, 9780471035688. 4. Nilesh B. Bahadure, Microprocessors - The 8086/8088, 80186/80286, 80386/80486 and the Pentium Family. 2010, PHI Learning, ISBN-978-81-203-3943-2. 5. Barry B. Brey, The INTEL Microprocessors – 8086 / 8088, 80186 / 80188, 80286, 80386, 80486 Pentium and Pentium pro processor, Pentium II, Pentium III and Pentium IV - Architecture, Programming and interfacing, PHI, 8th Edition, ISBN 0-13-502645-8.	

SKILL ENHANCEMENT COURSES

Year	I	**Course Code: CS1SE01E1**	**Credits**	3
Sem.	I	**Course Title: PYTHON PROGRAMMING**	**Hours**	60
Course Prerequisites, if any	Basic Knowledge in Programming Concepts			
Internal Assessment Marks: 50	End Semester Marks: 50	Duration of ESA (Theory) : 03 hrs. Duration of ESA (Practical) : 03 hrs.		
Course Outcomes	• Understand the basics of writing Python code • Implement programs using lists, tuples and dictionaries • Understand the use of control structures • Ability to write programs using packages • Understand the file manipulation			
Unit No.	**Course Content**		**Hours**	
	Theory Component			
Unit I	**Introduction, Data types** Introduction to Python – Advantages of using Python – Executing Python Programs – Python's Core data types – Numeric Types – String Fundamentals.		6	
Unit II	**Lists, Tuples, Dictionaries** Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing – list comprehension.		6	
Unit III	**Control Flow, Functions, Modules** Python Statements: Assignments – Expressions – If condition – While and For Loops. Functions: Definition, Calls – Scopes – Arguments – Recursive Functions– Functional Programming tools. Classes and Object-Oriented programming with Python. Modules and Packages: Purpose, using packages– Exception Handling with Python.		6	

Unit IV	**Packages** Packages: NumPy, Pandas, Scikit learn - Machine learning with Python – Cleaning up, Wrangling, Analysis, Visualization - Matplotlib package – Plotting Graphs.	6
Unit V	**File Handling** Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions	6
Practical Component		
Exercises	1. Exchange the values of two variables 2. Finding minimum among n variables 3. Perform Simple sorting 4. Generate Students marks statement 5. Find square root, GCD, exponentiation 6. Sum the array of numbers 7. Perform linear search, binary search 8. Perform Matrix operations using NumPy 9. Perform Dataframe operations using Pandas 10. Use Matplotlib on dataset and visualise 11. Perform Word count, copy file operations	30
Recommended Learning Resources		
Print Resources	1. Mark Lutz, “Learning Python”, Fifth Edition, O’Reilly, 2013. 2. Daniel Liang, “Introduction to programming using Python”, Pearson, First edition, 2021. 3. Wes Mc Kinney, “Python for Data Analysis”, O’Reilly Media, 2012. 4. Tim Hall and J-P Stacey, “Python 3 for Absolute Beginners”, Apress, First Edition, 2009. 5. Magnus Lie Hetland, “Beginning Python: From Novice to Professional”, Apress, Second Edition, 2005.	

Year	I	Course Code: CS1SE01E2	Credits	3
Sem.	I	Course Title: R Programming	Hours	60
Course Prerequisites, if any	NA			
Internal Assessment Marks: 50	End Semester Marks: 50	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Learn the basics in R programming • Understand to accessing variables and managing subsets of data • Design simple applications using the functions of R programming • Analyse the performance of the plotting tools in R programming • Create a project using the Lattice Package in R programming			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	Introducing to R – Script code – Graphing Facilities in R – Packages – General Issues in R – Getting Data into R – Importing Data.		6	
Unit II	Accessing variables and managing subsets of data – Accessing variables from a Dataframe – Accessing Subsets of Data – Combining Two Datasets with a Common Identifier – Exporting Data – Recoding Categorical Variables.		6	
Unit III	Simple Functions – The *tapply* Function – The *sapp*ly and *lapply* Functions – The summary Function – The table Function.		6	
Unit IV	An Introduction to Basic Plotting Tools – Adding a Smoothing Line – Loops and Functions – Introduction to Loops – Functions – More on Function and the if Statement - Graphing Tools – The Pie Chart – The Bar Chart and Strip Chart – Boxplot – Cleveland Dotplots – Revisiting the plot Function – The Pairplot – The Coplot – Combining Types of Plots.		6	
Unit V	An Introduction to the Lattice Package – High-level Lattice Functions – Multipanel Scatterplots: xyplot – Multipanel Boxplots: bwplot – Multipanel Cleveland Dotplots: dotplot – Multipanel Histograms: Histogram – Panel Functions – 3-D Scatterplots and Surface and Contour Plots – Common R mistakes.		6	
Practical Component				
Exercises	1. Data Import and Exploration: Read data, explore structure using head(), summary(), and str(). 2. Data Cleaning: Handle missing values, remove duplicates with duplicated(). 3. Data Visualization: Create plots (scatter, line, bar) using ggplot2. 4. Statistical Analysis: Conduct t-tests, chi-square tests, and ANOVA, perform correlation and regression. 5. Time Series Analysis: Analyze time series data, apply moving averages and smoothing. 6. Machine Learning: Train models with caret or randomForest, evaluate performance. 7. Text Analysis: Tokenize, preprocess text, perform sentiment analysis, and create word clouds.		30	

	8. Web Scraping: Use rvest or httr for web scraping, extract information. 9. Interactive Dashboards: Develop Shiny dashboards with widgets and plots. 10. Geospatial Analysis: Use sf and leaflet for geospatial visualization, perform spatial analysis. 11. Selection and Iteration: Write if statements and loops for decision-making and iteration. 12. 1D and 2D Arrays: Perform operations on vectors and matrices, calculate sums and means. 13. File Manipulation: Read and write files with functions like read.csv() and write.csv(). 14. Simulation and Randomization: Generate random numbers, simulate processes using runif() and rnorm(), use bootstrapping. 15. Function Writing: Write custom functions for specific tasks. 16. Interfaces with Other Languages: Interface R with C/C++ using Rcpp, call Python code with reticulate. 17. Package Development: Create and publish R packages, understand package structure and documentation.	
Recommended Learning Resources		
Print Resources	1. Alain F. Zuur, "A Beginner's Guide to R", Springer, 2019. 2. Norman Mat off, "The Art of R Programming: A Tour of Statistical Software Design", No Starch Press, 2011. 3. Jared P. Lander, "R for Everyone: Advanced Analytics and Graphics", Addison-Wesley Data & Analytics Series, 2013. 4. Mark Gardener, "Beginning R – The Statistical Programming Language", Wiley, 2013 5. Robert Knell, "Introductory R: A Beginner's Guide to Data Visualisation, Statistical Analysis and Programming in R", Amazon Digital South Asia Services Inc, 2013.	

SEMESTER II

Year	I	Course Code: CS2MJ02	Credits	4
Sem.	II	Course Title: Problem Solving & Programming Fundamentals	Hours	75
Course Prerequisites, if any	NIL			
Internal Assessment Marks: 25	End Semester Marks: 75	Duration of ESA (Theory) : 03 hrs. Duration of ESA (Practical) : 03 hrs.		
Course Outcomes	• Understand the basic concepts of programming languages, including syntax and semantics. • Apply programming constructs like loops, conditionals, and functions in practical scenarios. • Analyse code to identify and fix errors using debugging techniques. • Create modular programs using functions and procedures, emphasizing good programming practices.			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction to Computer Problem-Solving** The Problem-solving Aspect - Top-down Design - Implementation of Algorithms - Program Verification - The Efficiency of Algorithms - The Analysis of Algorithms.		9	
Unit II	**Basic programming constructs** Basic Data types (Numerical, String) – Variables – Expressions – I/O statements – Compile and Run - Debugging.		9	
Unit III	**Decision Making – Branching & Looping** Decision making – Relational Operators - Conditional statement, Looping statement - Nested loops - Infinite loops - Switch statements.		9	
Unit IV	**Array Techniques** Array Manipulation - Different operations - one dimensional array - two-dimensional array - multi-dimensional array - Character Arrays and Strings.		9	
Unit V	**Modular solutions** Introduction to functions – Importance of design of functions – Arguments – Parameters – return values – local and global scope – Recursion.		9	
Practical Component				
Exercises	1. Program to array counting, array order reversal & find the maximum number in a set. 2. Program for removal of duplicates from an ordered array & to partition an array. 3. Program to find the k^{th} smallest element. 4. Program to exchange the values of two variables without using a third variable. 5. Program that takes a list of numbers as input and counts the total number of elements in the list. 6. Program to calculate the sum of a set of numbers entered by the user. 7. Program to compute the factorial of a given integer. 8. Program to compute the sine of an angle (in degrees) using a series expansion.		30	

	9. Program to generate the Fibonacci sequence up to a specified limit. 10. Program that takes an integer as input and reverses its digits. 11. Program that converts a number from one base to another (e.g., binary to decimal, decimal to binary).	
Recommended Learning Resources		
Print Resources	1. R. G. Dromey, "How to solve it by Computer", Pearson Education, 2007. 2. E. Balaguruswamy, "Programming In ANSI C", 4th edition, TMH Publications, 2007. 3. Yashwant Kanetkar, "Let Us C", 13th Edition, PHP, 2013. 4. Allen B. Downey, "Think Python: How to Think like a Computer Scientist", 2nd Edition, O'Reilly Publishers, 2016.	

Year	I	Course Code: CS2MI02	Credits	4
Sem.	I	Course Title: Microcontroller Programming	Hours	75
Course Prerequisites, if any	Digital Logic Fundamentals, Microprocessor and ALP			
Internal Assessment Marks: 25	End Semester Marks: 75	Duration of ESA (Theory) : 03 hrs. Duration of ESA (Practical) : 03 hrs.		
Course Outcomes	• Learn the fundamentals of Microcontrollers. • Understand the internal design of 8051 microcontroller along with the features and their programming. • Analyse the on-chip peripherals of microcontrollers. • Design different interfacing applications using microcontrollers and peripherals. • Build systems using microcontrollers for real time applications.			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Microprocessors and Microcontrollers** Microprocessors vs Microcontrollers - 8051 Architecture - Input/Output Pins – Ports - External Memory - Counter and Timers - Serial Data Input/Output - Interrupts.	9
Unit II	**Programming 8051** Addressing Modes, External Data Moves, Code Memory Read-Only Data Moves, PUSH and POP Opcodes, Data Exchanges - Logical Operations - Arithmetic Operations - Jump and Call Opcodes.	9
Unit III	**8051 Microcontroller Design** Microcontroller Specification – Design - Testing - Timing Subroutines - Lookup Tables for 8051 - Serial Data Transmission.	9
Unit IV	**Applications** Keyboards – Displays - Pulse Measurement - D/A and A/D Conversions - Multiple Interrupts.	9
Unit V	**Serial Data Communication** Network Configurations - 8051 Data Communication Modes.	9
	Practical Component	
Exercises	1. Blinking LED 2. Reading Analog Input 3. Digital Counter with Seven-Segment Display 4. Analog-to-Digital Conversion (ADC)	30

	5. UART Communication 6. PWM (Pulse Width Modulation) Control 7. Timer Interrupt - Using a timer interrupt to perform a task at regular intervals 8. I2C Communication 9. External Interrupt 10. Temperature Sensor (DS18B20) Interface 11. Matrix Keypad Interface 12. LCD Display Interface 13. Traffic Light Controller (Many more programs can be included related to the programming 8051 microcontroller)	
Recommended Learning Resources		
Print Resources	1. Kenneth J. Ayala, The 8051 Microcontroller Architecture, Programming, and Applications, West Publishing Company, USA, 1991. 2. Martin Bates, PIC Microcontrollers - An Introduction to Microelectronics, Third Edition, Elsevier, 2011, ISBN: 978-0-08-0969 1 1-4 3. Hubert Henry Ward, C Programming for the PIC Microcontroller-Demystify Coding with Embedded Programming, Apress, UK, 2020. ISBN-13 (pbk): 978-1-4842-5524-7; ISBN-13 (electronic): 978-1-4842-5525-4 https://doi.org/10.1007/978-1-4842-5525-4	

SKILL ENHANCEMENT COURSES

Year	I	**Course Code: CS1SE02E1**	**Credits**	3
Sem.	II	**Course Title: PROGRAMMING FOR MOBILE DEVICES**	**Hours**	60
Course Prerequisites, if any	Basic computer programming skill			
Internal Assessment Marks: 50	End Semester Marks: 50	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the basics of Android Ecosystem. • Learn to use Android Ecosystem. • Design simple applications using the programming constructs in Kotlin. • Analyse the process of building interactive apps, Games, Social Media apps. • Create applications for TVs, Wearable and Android Auto.			

Unit No.	**Course Content**	**Hours**
	Theory Component	
Unit I	**Introduction** About the Android Ecosystem - Installing Software tools - Creating an Android App - Examining a basic Android app - Improving the App.	6
Unit II	**Android Background Material** Using Android Studio - Kotlin for Java programmers - Kotlin for Everyone - Object Orientation in Kotlin - Functional Programming in Kotlin - An Introduction to XML.	6
Unit III	**The Building Blocks:** Overview of Jetpack - Building foundations for the App - Architecture of the App - Defining App's behaviour - Interactivity.	6
Unit IV	**Adding Cool Features** Building a Game in Android - Case study of building a Social Media App - Building Native applications	6
Unit V	**Apps for Tablets, Watches, TVs and Cars** Apps for Tablets - Developing for Android Wear - Developing Android TV apps - Case study of App building with Android Auto.	6

Practical Component		
Exercises	1. Configure Android Studio and set the development environment. 2. Build a basic Android app for numerical calculations. 3. Build an Android app to use various sensors of the device. 4. Build a simple gaming app. 5. Build a social media app. 6. Case study : Build a calendar for Tablets. 7. Build an app for the Android Wear. 8. Build a simple app for Android Auto. 9. Build an Android TV app. 10. Case Study : Native Apps.	30
Recommended Learning Resources		
Print Resources	1. Barry Burd, John Paul Mueller, Android Application Development All-in-one for Dummies, Third Edition (2021) Wiley India, ISBN: 978-9354245787. 2. Dawn Griffiths, David Griffiths, Head First Android Development: A Learner's Guide to Building Android Apps with Kotlin, Third Edition (2021), O'Reilly, ISBN: 978-9355420855.	

Year	I	Course Code: CS1SE02E2	Credits	3
Sem.	II	Course Title: Visual Programming using C#	Hours	60
Internal Assessment Marks: 50	End Semester Marks: 50	Duration of ESA (Theory) : 03 hrs. Duration of ESA (Practical) : 03 hrs.		
Course Outcomes	• Understand the key components of the .NET Framework related to C# development. • Learn the basic syntax and structure of C# programs. • Design C# applications by integrating various object-oriented programming techniques in .NET framework. • Analyze the significance of graphical user interface (GUI) components and the Event Handling Model using C# programming. • Create robust, scalable database applications using ADO.NET connectivity.			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction to .Net Framework** An Overview - Framework Components - The Common Language Runtime (CLR) - .NET Base Class Library - Common Language Specification (CLS) - Common Type System (CTS) - Metadata and Assemblies - .NET Namespaces - MSIL - JIT Compilers.	6
Unit II	**Overview of C#** Program structure, Literals, Variables, Constants, Data Types, Operators, Statements and Expressions, Branching, Looping and loop control statements, Arrays, Strings manipulation, Boxing and Unboxing, Pre-processors, Namespaces.	6
Unit III	**Object Oriented Programming in C#** Class, Objects, Encapsulation, Constructors and its types, Inheritance, Polymorphism. Interface, Abstract class, Operator overloading, Properties, Indexers, Delegates, Collections.	6
Unit IV	**Windows Forms** Introduction to Windows Forms and various controls, SDI and MDI applications, Menu Creation, Common Dialog Boxes. Events and event handling.	6
Unit V	**Introduction to ADO.NET** ADO.NET Architecture - Connection Object - Command Object - Dataset - Data Reader Object - Data Adapter Object- Data Table - Datagridview and Data Binding. Connecting to a database and OLE DB data source, Adding, updating, deleting, and viewing records in the database.	6
	Practical Component	
Exercises	1. Installation of Visual Studio and creation of Simple console Application. 2. Create simple C# program for the following concepts: a. To Check whether the given number is Prime or not b. To Check whether the given number is Armstrong or not c. To demonstrate Pascal's Triangle d. To Check whether the alphabet is a vowel or not using switch..case. e. To Check whether the given string is palindrome or not using arrays.	30

	3. Create a program to demonstrate boxing and unboxing operations. 4. Implement Classes and Objects, Inheritance & Polymorphism 5. Implement Interfaces and Operator Overloading. 6. Create a GUI using standard controls, SDI & MDI forms. 7. Design an application with menu options and Common Dialog box. 8. Design a database application using Database Controls 9. Design an ADO database and perform the operations of insertion, modification, deletion, and viewing. 10. Develop any TWO case studies listed below: a. Inventory Control b. Retail Shop Management c. Employee Information System d. Personal Assistant Program e. Students' Information System	
Recommended Learning Resources		
Print Resources	1. Herbert Schildt, "The Complete Reference: C# 4.0", Tata McGraw Hill, 2012. 2. Christian Nagel et al. "Professional C# 2012 with .NET 4.5", Wiley India, 2012. 3. Andrew Troelsen, "Pro C# 2010 and the .NET 4 Platform", Fifth edition, A Press, 2010. 4. Ian Griffiths, Matthew Adams, Jesse Liberty, "Programming C# 4.0", Sixth Edition, O'Reilly, 2010.	

VALUE ADDED COURSES

Year	I	Course Code: CS2VA04	Credits	2
Sem.	II	Course Title: Digital Technologies	Hours	45
Course Prerequisites, if any	-NIL-			
Internal Assessment Marks: 25	End Semester Marks: 75	Duration of ESA (Theory) : 03 hrs. Duration of ESA (Practical) : 03 hrs.		
Course Outcomes	• Learn about digital paradigm. • Understand the importance of digital technology, digital financial tools, e-commerce. • Analyse the concepts of communication and networks. • Understand the e-governance and Digital India initiatives. • Understand the use & applications of digital technology. • Learn the applications of machine learning and big data.			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	Introduction & Evolution of Digital Systems. Role & Significance of Digital Technology. Information & Communication Technology & Tools. Computer System & its working, Software and its types. Operating Systems: Types and Functions. Problem Solving: Algorithms and Flowcharts.		7	
Unit II	Communication Systems: Principles, Model & Transmission Media. Computer Networks & Internet: Concepts & Applications, WWW, Web Browsers, Search Engines, Messaging, Email, Social Networking. Computer Based Information System: Significance & Types. E-commerce & Digital Marketing: Basic Concepts, Benefits & Challenges.		7	
Unit III	Digital India & e-Governance: Initiatives, Infrastructure, Services and Empowerment. Digital Financial Tools: Unified Payment Interface, Aadhar Enabled Payment System, USSD, Credit / Debit Cards, e-Wallets, Internet Banking, NEFT/RTGS and IMPS, Online Bill Payments and PoS. Cyber Security: Threats, Significance, Challenges, Precautions, Safety Measures, & Tools, legal and ethical perspectives.		7	
Unit IV	Emerging Technologies & their applications: Overview of Cloud Computing, Big Data, Internet of Things, Virtual Reality,		7	
Unit V	Emerging Technologies & their applications: Blockchain & Cryptocurrency, Robotics, Machine Learning & Artificial Intelligence, 3-D Printing. Digital Signatures.		7	
Practical Component				
Practice	1. Operating System Installation and configuration 2. Application Software Installation and configuration 3. Hardware understanding and minor troubleshooting 4. Networking, cabling, configuration		10	
Recommended Learning Resources				
Print Resources	1. Pramod Kumar, Anuradha Tomar, R. Sharmila, "Emerging Technologies in Computing - Theory, Practice, and Advances", Chapman and Hall / CRC, 1st Edition, 2021, eBook ISBN: 9781003121466. https://doi.org/10.1201/9781003121466. 2. V. Rajaraman, "Introduction to Information Technology", PHI, 3rd Edition, 2018, ISBN-10: 9387472299, ISBN-13: 978-9387472297.			

	3. E. Balagurusamy, “Fundamentals of Computers”, Tata Mc GrawHill, 2nd Edition, 2011, ISBN: 9780071077880. 4. Behrouz A. Forouzan, “Data Communications and Networking”, McGraw Hill, 4th Edition, 2007, ISBN 978-0-07-296775-3. 5. Rajkumar Buvya, James Broberg, and Andrzej Gosciniski, “Cloud Computing-Principals and Paradigms”, Wiley, 2011, ISBN: 978-0-470-88799-8. 6. Stuart Russel and Peter Norvig, “Artificial Intelligence - A Modern Approach”, Pearson Education, 3rd Edition, 2010, ISBN- 13: 978-0-13 -604259-4. 7. Samuel Greengard, “Internet of Things”, The MIT Press, 2015, ISBN: 9780262328937, https://doi.org/10.7551/mitpress/10277.001.0001. 8. C.S.V. Murthy, “E- Commerce – Concept, Models &Strategies”, Himalaya Publishing House, 2015, ISBN: 8178662760. 9. Hurwith, Nugent Halper, Kaufman, “Big Data for Dummies”, Wiley & Sons, 1st Edition, 2013, ISBN-13: 978-1118504222.