THIRUVALLUVAR UNIVERSITY

SERKKADU, VELLORE–632115

B.SC. COMPUTER SCIENCE

SYLLABUS

(University Department)

FROM THE ACADEMIC YEAR

2024 – 2025

Bachelor of Science in Computer Science

Choice Based Credit System (CBCS) with
Learning Outcome based Curriculum Framework (LOCF) Norms

Syllabus

1.About the Program

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 Thiruvalluvar 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 Objectives

The B.Sc. Computer Science Programme is designed with the following specific objectives.

i. To ensure students grasp fundamental concepts in computer science such as algorithms, data structures, programming paradigms, and computational thinking.
ii. To develop proficiency in programming languages commonly used in the field, as well as understanding software development methodologies and tools.
iii. To cultivate problem-solving skills through algorithmic thinking and applying appropriate data structures and algorithms to solve computational problems efficiently.
iv. To provide an understanding of computer architecture, operating systems, networks, and other foundational aspects of computer systems.

3. Admission Eligibility

Students who have cleared their class 12 in the Science stream with PCM subjects (Physics, Chemistry, and Mathematics) can be admitted to a BSc. Computer Science in a college. Along with PCM subjects, candidates should have passed with a minimum of marks from a recognized board.

4. Duration of Programme

The programme shall normally extend over a period of three academic years consisting of six semesters.

5. Programme Beneficial

Computer science encourages creativity and innovation. Through programming and designing algorithms, you can create new software, applications, or even hardware solutions that can make a significant impact on society.

A degree or certification in computer science can open up a wide range of career opportunities. From software developer to data scientist to IT consultant, there are numerous paths you can pursue within the field.

Studying computer science can also help you understand the societal and ethical implications of technology. From privacy concerns to the impact of automation on jobs, a background in computer science can provide valuable insights into these issues.

6. B.Sc.,(Computer Science) Career Opportunity

- **Software Developer**: Design, develop, and maintain software applications for various platforms such as web, mobile, desktop, and embedded systems. This role involves programming, problem-solving, and collaborating with cross-functional teams.
- **Systems Analyst**: Analyze and improve computer systems and processes within organizations to enhance efficiency, productivity, and alignment with business goals. Systems analysts bridge the gap between IT and business, identifying technological solutions to meet organizational needs.
- **Network Administrator**: Design, implement, and maintain computer networks to ensure seamless communication and data exchange within organizations. Network administrators/engineers are responsible for network security, performance optimization, and troubleshooting.
- **Cybersecurity Specialist**: Protect computer systems, networks, and data from cyber threats by implementing security measures, conducting vulnerability assessments, and responding to security incidents. Cybersecurity specialists play a crucial role in safeguarding sensitive information and preventing cyber attacks.
- **Technical Support Specialist**: Provide technical assistance and troubleshooting support to users experiencing software, hardware, or network problems. Technical support specialists diagnose issues, provide solutions, and escalate complex problems to higher-level support teams when necessary.
- **Game Developer**: Design, develop, and deploy video games for various platforms, including consoles, PCs, and mobile devices. Game developers create game mechanics, implement features, and optimize performance to deliver immersive gaming experiences.

7. Learning Outcomes-Based Curriculum Framework Guidelines Based Regulations For Under Graduate Programme

Programme:	**B.Sc., Computer Science**
Programme Code:	
Duration:	**3 Years of Under Graduates**
Program Outcome	**On Successful completion of the Program the students will be able to** **PO1:** An ability to apply knowledge of computing and mathematics appropriate to the discipline. **PO2:** An ability to analyse a problem, and identify and define the computing requirements appropriate to its solution. **PO3:** An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs. **PO4:** An ability to function effectively on teams to accomplish a common goal. **PO5:** An understanding of professional, ethical, legal, security, and social issues and responsibilities. **PO6:** An ability to communicate effectively with a range of audiences.
Programme Specific Outcomes:	**On Successful completion of Bachelor of Science in Computer Science with Cognitive Systems Programme, the students should be able to:** **PSO1:** An ability to analyse, design, develop, and maintain software systems using appropriate software techniques and tools. **PSO2:** An ability to design and implement computer algorithms and data structures for solving complex computational problems. **PSO3:** An ability to apply principles of computer networking and distributed systems to design and manage networked systems and applications. **PSO4**: An ability to design, implement, and manage database systems to efficiently store, retrieve, and manipulate data . **PSO5:** An ability to analyse, design, and develop intelligent systems using principles of artificial intelligence and machine learning. **PSO6:** An ability to apply principles of cybersecurity to design and implement secure computing systems and protect against cyber threats.

PO/PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
PO1	✓	✓				
PO2	✓			✓	✓	
PO3			✓	✓		
PO4	✓				✓	✓
PO5				✓		✓
PO6		✓	✓			✓

8. Teaching and Assessment Methodology

A. The teaching methodology adopted for the course will utilize participatory learning methods, like workshops, discussions, assignments, short education tours, seminars, peer teaching, and group work, apart from regular lectures.
B. The syllabus indicates the type of teaching method, to be adopted for a particular topic, in the footnote of the same page.
C. The method suggested is only indicative; the concerned course teacher can use other methods or a combination of many methods, in order to improve the quality of knowledge transfer.
D. Course teachers adopting participatory teaching methods may please take extra care on the following issues
    a. Set a brief, clear task rather than lecturing
    b. Usehands-on,multi-sensorymaterialsratherthanrelyonlyonverbalcommunication
    c. Create an informal, relaxed atmosphere
    d. Choose growth-producing activities Evoke feelings, beliefs, needs, doubts, perceptions, aspirations
    e. Encourage creativity, analysis, planning
    f. Decentralize decision-making
E. The following portions give details of some contemporary techniques that may be followed by course teachers, who teach various subjects in criminology.

1. BRAINSTORMING

Brainstorming is a familiar technique in which the teacher asks a specific question or describes a particular scenario, and students offer many different ideas. These ideas are then usually written on a flipchart or chalkboard and considered for further discussion.

2. CASEANALYSIS

A case study is a written scenario that usually involves an important community situation. Since it is written beforehand, it can be specifically created to address relevant local issues.

3. DEMONSTRATIONS/PRACTICALEXPOSURE

A demonstration is a structured performance of an activity to show, rather than simply tell, a group how the activity is done. This method brings to life some information that you may have already presented in a lecture.

4. DRAMATIZATION

A dramatization is a carefully scripted play where the characters act out a scene related to a learning situation. It is designed to bring out the important issues to be discussed or messages to be learned.

5. FISHBOWL

In a fishbowl discussion, most of the students sit in a large circle, while a smaller group of students sits inside the circle. The fishbowl can be used in two distinct ways:

- As a structured brainstorming session: Choose a specific topic based on the group's needs or interests. A handful of seats are placed inside a larger circle. Students who have something to say about the topic at hand sit in the center. Anyone sitting inside the fishbowl can make a comment, offer information, respond to someone else in the center, or ask a question. When someone from the outside circle has a point to make, he or she taps the shoulder of someone in the center and takes that person's seat. This continues, with people from the outside tapping and replacing people on the inside, as a lively brainstorm takes place. You will need to process the many ideas after the fishbowl exercise.
- For structured observation of a group process: Students in the fishbowl are given a specific task to do, while students outside the fishbowl act as observers of the group process. The inner group works on its task together, and the outer group is asked to note specific behaviors. To process the activity, ask the inner group to reflect on the group process, and ask the outer group to describe what they observed.

6. GAMES

Games are appropriate participatory tools when they are used to encourage students to take charge of their own learning, and to test and reinforce new knowledge or skills. Adapt a popular game to convey or test knowledge of a particular topic, or create a new game to test or reinforce learning. Divide students into groups, if necessary, to play the game. Use games after information has already been shared using another method (e.g., lecturer, demonstration, jigsaw learning, etc.) or to assess students' knowledge at the start of a learning activity.

7. JIGSAWLEARNING

In a jigsaw activity, evenly divided groups are given a topic to learn(a piece of the puzzle to master). Once these small groups have mastered the content, the groups are reorganized so that each new group contains one member from each original group (now each group contains all essential pieces of the puzzle to put together). Each new group now contains an "expert" on the content that they have mastered in the original groups, and one at a time, each expert teaches the new content to the newly formed groups. The teacher then processes the activity and emphasizes key learning.

8. LECTURETTE

A lecturette is a short, oral presentation of facts or theory. No more than 15-20 minutes in length, the goal of a lecturette is to impart information in a direct, highly organized fashion. The course teacher presents knowledge on a topic, sometimes using flipcharts, computer software presentations or other media to guide the discussion.

9. PANELDISCUSSIONS

This method usually involves the presentation of an issue by several teachers at a table in front of a group. Usually, each teacher speaks briefly on the topic and then a moderator solicits questions from the audience. The moderator introduces the presenters/ teachers, keeps the discussion on the topic and within time limits and summarizes the discussion at the end. Each teacher typically speaks for a set period of time (for example, five minutes). After all teachers have spoken, the moderator invites questions from students. At the end of the session, the moderator may summarize the discussion and thank the presenters for their participation.

10. ROLEPLAY

Role-plays are short interactions of students playing specific, predetermined roles to explore issues or practice skills. Roles are usually written out, and the teacher may help students playing the roles understand "who" they are to be. Role-plays are generally used after a period of instruction or discussion. For example, if students are learning communication skills, groups can role-play being assertive in typical situations (e.g., students in peer pressure situations, or people needing to access services in a clinic or office). Stop the role-play periodically and discuss what behaviors worked and what was difficult and allow the group to brainstorm different choices of behavior/words. The role-play may be done again, with the same person practicing or someone else trying.

11. SKIT

A skit is an impromptu performance by students to demonstrate something they know. Skits can be created by students to show concerns they have about such things as peer pressure, victim issues in their community or lack of resources. Give students a topic, the maximum length of the skit and the amount of time they have to prepare (depending on the complexity, 30 minutes or an afternoon, for example).

12. SMALL GROUP DISCUSSION

A small group discussion is a structured session in which three to six students exchange ideas and opinions about a particular topic or accomplish a task together. After the groups have had an opportunity to work together, they report the highlights of their work back to the large group, and the teacher helps the group process the activity. Begin the learning activity by briefly presenting a topic to the large group. Then, divide the group into smaller groups and set a clear task for the small groups to accomplish. Write directions, goals and time allotted for the task on a chalkboard, flipchart or handout. As groups are working, walk around and listen in briefly to each group. Keep groups focused by announcing the time remaining periodically. After the small group work, students typically reassemble in the large group and a representative from each small group shares their findings to the large group for a whole group discussion. Help the group process the activity to be sure the intended message was conveyed.

9. Credit Distribution for UG Programme in Computer Science

B.Sc. Computer Science

FIRST YEAR

Semester-I

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-I	Language	3	6
Part-II	English	3	6
Part-III	Core Courses 2 (CC1, CC2)	10	11
	Elective Course 1 (Generic / Discipline Specific)EC1	3	5
Part-IV	Skill Enhancement Course SEC-1 (Non Major Elective)	2	2
	Foundation Course FC	2	2
	Total	**23**	**32**

Semester-II

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-I	Language	3	6
Part-II	English	3	6
Part-III	Core Courses 2 (CC3, CC4)	10	10
	Elective Course 1 (Generic / Discipline Specific) EC2	3	6
Part-IV	Skill Enhancement Course -SEC-2 (Non Major Elective)	2	2
	Skill Enhancement Course -SEC-3 (Discipline Specific / Generic)	2	2
	Total	**23**	**32**

SECOND YEAR

Semester-III

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-I	Language	3	6
Part-II	English	3	6
Part-III	Core Courses 2 (CC5, CC6)	10	10
	Elective Course 1 (Generic / Discipline Specific)EC3	3	5
Part-IV	Skill Enhancement Course -SEC-4 (Entrepreneurial Based)	1	1
	Skill Enhancement Course -SEC-5 (Discipline Specific/ Generic)	2	2
	Environmental Studies(EVS)	2	2
	Total	**24**	**32**

Semester-IV

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-I	Language	3	6
Part-II	English	3	6
Part-III	Core Courses 2 (CC7, CC8)	10	10
	CC7: Core Industry Module -1 - **Industrial Statistics**		
	CC8 : Any Core paper		
	Elective Course 1 (Generic / Discipline Specific)EC4	3	4
Part-IV	Skill Enhancement Course -SEC7	2	2
	Skill Enhancement Course -SEC-8 (Discipline Specific / Generic)	2	2
	Total	**25**	**32**

THIRD YEAR

Semester-V

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-III	Core Courses 3(CC9, CC10, CC11)	12	15
	Elective Courses 2 (Generic / Discipline Specific) EC5, EC6	6	8
	Core /Project with Viva voce CC12	4	5
Part-IV	Value Education	2	2
	Internship / Industrial Training (Carried out in II Year Summer vacation) (30 hours)	2	0
	Total	**26**	**30**

Semester-VI

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-III	Core Courses 3 (CC13, CC14, CC15)	12	18
	Elective Courses 2 (Generic / Discipline Specific) EC7, EC8	6	10
Part IV	Professional Competency Skill Enhancement Course SE8	2	2
Part-V	Extension Activity (Outside college hours)	1	-
	Total	**21**	**30**

Total Credits: 140

10. Credit Distribution for UG Programme

Sem I	Credit	H	Sem II	Credit	H	Sem III	Credit	H	Sem IV	Credit	H	Sem V	Credit	H	Sem VI	Credit	H
1.1. Language	3	6	2.1. Language	3	6	3.1. Language	3	6	4.1. Language	3	6	5.1 Core Course – \CC IX	4	5	6.1 Core Course – CC XIII	4	6
1.2 English	3	6	2.2 English	3	6	3.2 English	3	6	4.2 English	3	6	5.2 Core Course – CC X	4	5	6.2 Core Course – CC XIV	4	6
1.3 Core Course – CC I	5	6	2.3 Core Course – CC III	5	5	3.3 Core Course – CC V	5	5	4.3 Core Course – CC VII Core Industry Module	5	5	5. 3.Core Course CC -XI	4	5	6.3 Core Course – CC XV	4	6
1.4 Core Course – CC II	5	5	2.4 Core Course – CC IV	5	5	3.4 Core Course – CC VI	5	5	4.4 Core Course – CC VIII	5	5	5. 3.Core Course –/ Project with viva-voce CC - XII	4	5	6.4 Elective -VII Generic/ Discipline Specific	3	5
1.5 Elective I Generic/ Discipline Specific	3	5	2.5 Elective II Generic/ Discipline Specific	3	6	3.5 Elective III Generic/ Discipline Specific	3	5	4.5 Elective IV Generic/ Discipline Specific	3	6	5.4 Elective V Generic/ Discipline Specific	3	4	6.5 Elective VIII Generic/ Discipline Specific	3	5
1.6 Skill Enhancement Course SEC-1 (NME)	2	2	2.6 Skill Enhancement Course SEC-2 (NME)	2	2	3.6 Skill Enhancement Course SEC-4, (Entrepreneurial Skill)	1	1	4.6 Skill Enhancement Course SEC-6	2	2	5.5 Elective VI Generic/ Discipline Specific	3	4	6.6 Extension Activity	1	-
1.7 Skill Enhancement -(Foundation Course)	2	2	2.7 Skill Enhancement Course – SEC-3	2	2	3.7 Skill Enhancement Course SEC-5	2	2	4.7 Skill Enhancement Course SEC-7	2	2	5.6 Value Education	2	2	6.7 Professional Competency Skill	2	2
						3.8 E.V.S	2	2				5.5 Summer Internship /Industrial Training	2	0			
Total	**23**	**32**	**Total**	**23**	**32**	**Total**	**24**	**32**	**Total**	**23**	**32**	**Total**	**26**	**30**	**Total**	**21**	**30**
Total Credit Points - 140																	

11. Template for Curriculum Design for UG Programme in Computer Science

Semester-I

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-I	Language	3	6
Part-II	English	3	6
Part-III	CC1 - Python Programming	5	6
	CC2 - Practical : Python Programming	5	5
	Elective Course 1 (Generic / Discipline Specific) – Discrete Mathematics	3	5
Part-IV	Skill Enhancement Course- SEC-1 (Non Major Elective)- Office Automation	2	2
	Foundation Course FC - Problem Solving Techniques	2	2
	Total	**23**	**32**

Semester-II

Part	List of Courses	Credit	Hours per week(L/T/P)
Part-I	Language	3	6
Part-II	English	3	6
Part-III	CC3 - Data Structure and Algorithms	5	5
	CC4 - Practical: Data Structure and Algorithms	5	5
	Elective Course 2 (Generic / Discipline Specific) – Graph Theory and its applications	3	6
Part-IV	Skill Enhancement Course- SEC-2 (Non Major Elective) - Quantitative Aptitude	2	2
	Skill Enhancement Course – SEC-3 (Discipline / Subject Specific) – Advanced Excel	2	2
	Total	**23**	**32**

Semester-III

Part	List of Courses	Credit	Hours per week(L/T/P)
Part-I	Language	3	6
Part-II	English	3	6
Part-III	CC5- Programming in Java	5	5
	CC6 - Practical: Programming in Java	5	5
	Elective Course 3 (Generic / Discipline Specific) - EC3 - Discrete Mathematical Structures	3	5
Part-IV	Skill Enhancement Course -SEC-4 (Entrepreneurial Based) – PHP Programming	1	1
	Skill Enhancement Course -SEC-5 (Discipline Specific/ Generic) – Scripting Language	2	2
	Environmental Studies	2	2
	Total	**24**	**32**

Semester-IV

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-I	Language	3	6
Part-II	English	3	6
Part-III	CC7 - .NET Programming	5	4
	CC8 - Practical: : .NET Programming	5	4
	Elective Course - EC4 (Generic / Discipline Specific) – Statistics Analysis using R	3	4
Part-IV	Skill Enhancement Course – SEC-6 -Cloud Computing	2	2
	Skill Enhancement Course - SEC-7 -Big Data Analytics	2	2
	Total	**23**	**32**

Semester-V

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-III	CC9 - Software Engineering	4	5
	CC10 - Database Management System	4	5
	CC11 - Practical: Database Management System	4	5
	Elective Course – EC5 (Generic / Discipline Specific) – Operating Systems	3	5
	Elective Course – EC6 (Generic / Discipline Specific) – Data Mining and Warehousing	3	4
	CC12 - Core /Project with Viva voce	4	4
Part-IV	Value Education	2	2
	Internship / Industrial Training (Summer vacation at the end of IV semester activity)	2	0
	Total	**26**	**30**

Semester-VI

Part	List of Courses	Credit	Hours per week (L/T/P)
Part-III	CC13 - Computer Networks	4	6
	CC14 –Artificial Intelligence	4	6
	CC15 - Practical: Artificial Intelligence	4	6
	Elective Course – EC7 (Generic / Discipline Specific) – Introduction to Data Science	3	5
	Elective Course – EC8 (Generic / Discipline Specific) – Cyber security	3	5
Part-IV	Professional Competency Skill Enhancement Course SE8	2	2
Part -V	Extension Activity	1	-
	Total	**21**	**30**

Total Credits : 140

Consolidated Semester wise and Component wise Credit Distributions

Part	Sem I	Sem II	Sem III	Sem IV	Sem V	Sem VI	Total Credits
Part I	3	3	3	3	-	-	12
Part II	3	3	3	3	-	-	12
Part III	13	13	13	13	22	18	92
Part IV	4	4	5	4	4	2	23
Part V	-	-	-	-	-	1	01
Total	23	23	24	23	26	21	140

*Part I. II , and Part III components will be separately taken into account for CGPA calculation and classification for the under graduate programme and the other components. IV, V have to be completed during the duration of the programme as per the norms, to be eligible for obtaining the UG degree.

Over all Subject and Credit Summary

Subject	Category	Papers	Credit (Max)	Total Credits	Marks	Total Marks
Language I(Tamil)	Part I	4	3	12	100	400
Language II(English)	Part II	4	3	12	100	400
Core **Theory**	Part III	8	5	36	100	800
Core **Practical**	Part III	6	5	28	100	600
Core **Elective**	Part III	8	3	24	100	800
Skill Enhancement Course **(SEC)**	Part IV	7	2	13	100	700
Foundation Course	Part IV	1	2	02	100	100
Core **Project**	Part III	1	4	04	100	100
Extension Activity	Part V	1	1	01	100	100
EVS& Value Education	Part IV	2	2	04	100	200
Professional Competency Skills	Part IV	1	2	02	100	100
Summer Internship/ Industrial Visit	Part IV	1	2	02	100	100
Total	**-**	**44**	**-**	**140**	**-**	**4400**

ANNEXURE I

S. No	Generic Electives	S. No	Discipline Specific
1	Discrete Mathematics – I	1	Analytics for Service Industry
2	Discrete Mathematics – II	2	Natural Language Processing
3	Numerical Methods – I	3	Grid Computing
4	Numerical Methods – II	4	
5	Mathematical Statistics – I	5	Data Communication And Computer Networks
6	Mathematical Statistics – II	6	Big Data Analytics
7	Electronics Science	7	Computer Networks
8	Nanotechnology	8	Cryptography
9	Optimization Technique / Operational Research	9	Operating System
10	Introduction to Linear Algebra	10	Artificial Neural Networks
11	Graph Theory and Its Applications	11	Software Engineering
12	Digital Logic Fundamentals	12	Distributed Computing
13	Microprocessor & Micro Controller And more.,	13	Analytics for Service Industry
		14	Agile Project Management
		15	Computing Intelligence
		16	Information Security And more.,

SKILL ENCHANCEMENT

1	INTRODUCTION TO HTML
2	OFFICE AUTOMATION
3	QUALITATIVE APTITUDE
4	CYBER FORENSICS
5	MULTIMEDIA SYSTEMS
6	SOFTWARE TESTING
7	DATA MINING AND WAREHOUSING
8	BIO METRICS
9	ENTERPRISE RETAIL PLANNING
10	WEB TECHNOLOGY
11	ROBOTICS AND APPLICATIONS
12	SIMULATION AND MODELING
13	PATTERN RECOGNITION
14	ADVANCED EXCEL
15	OPEN SOURCE SOFTWARE TECHNOLOGIES
16	PHP PROGRAMMING and more..,

SYLLABUS

First Year (Semester – I)

Course Code-1	**Python Programming**		**Credits 5**
Lecture Hours:(L) Per week - 4	**TutorialHours:75 (T)per week**	**Lab Practice Hours: (P)per week**	**Total:(L+T +P) Per week: 4**
Course Category: Core	**Year & Semester: I Year I Semester**	**Admission Year:**	
Pre-requisite	Basic Knowledge of Programming concept		

LearningObjectives:(forteachers:whattheyhavetodointheclass/lab/field)

- Describe the core syntax and semantics of Python programming language.
- Discover the need for working with the strings and functions.
- Illustrate the process of structuring the data using lists, dictionaries, tuples and sets.
- Understand the usage of packages and Dictionaries

Course Outcomes	**Program Outcomes**
CO1: To Basic concept of Python Programming.	PO1,PO2,PO3, PO4,PO5,PO6
CO2: To learn simple Python programs using conditionals and looping for solving problems.	PO1,PO2,PO3, PO4,PO5,PO6
CO3: To Understand the basics python functions and arguments.	PO1,PO2,PO3, PO4,PO5,PO6
CO4: To develop the knowledge of python programming Object and their uses.	PO1,PO2,PO3, PO4,PO5,PO6
CO5: To understand the python packages.	PO1,PO2,PO3, PO4,PO5,PO6

Recap:(notforexamination)Motivation/previouslecture/relevantportionsrequiredforthe course)[Thisisdoneduring2Tutorialhours)

Units	**Contents**	**Required Hours**
I	**Basics of Python Programming:** History of Python-Features of Python-Literal-Constants-Variables-Identifiers–Keywords-Built-in Data Types-Output Statements–Input Statements-Comments – Indentation- Operators-Expressions-Type conversions.**Python Programming Class & Object**: Class and Object-Abstraction-Ploymorphism- Inheritance .	**15**
II	**Strings:** String operations- Immutable Strings - Built-in String Methods and Functions. **Control Statements:** Selection/Conditional Branching statements: if, if-else, nested if and if-elif-else statements. **Iterative Statements:** while loop, for loop, else suite in loop and nested loops. Jump Statements: break, continue and pass statements.	**15**

III	**Lists:** Creating a list -Access values in List-Updating values in Lists-Nested lists - Basic list operations-List Methods. **Tuples:** Creating, Accessing, Updating and Deleting Elements in a tuple – Nested tuples– Difference between lists and tuples. **Dictionaries:** Creating, Accessing, Updating and Deleting Elements in a Dictionary – Dictionary Functions and Methods - Difference between Lists and Dictionaries.	**15**
IV	**Functions:** Function Definition – Function Call – Variable Scope and its Lifetime-Return Statement. **Function Arguments:** Required Arguments, Keyword Arguments, Default Arguments and Variable Length Arguments- **Recursion**:- Regular Expressions – Concept of regular expression, various types of regular expressions, using match function. **Modules:** import statement- The Python module – dir() function – Modules and Namespace – Defining our own modules.	**15**
V	**File Handling:** Types of files in Python - Opening and Closing files-Reading and Writing files: write() and writelines() methods- append() method – read() and readlines() methods – with keyword – Splitting words – File methods - File PositionsRenaming and deleting files. Exception-Error Handling.	**15**
	Total Hours	**75**

Learning Resources:

- **Recommended Texts**
  1. Reema Thareja, "Python Programming using problem solving approach", First Edition, 2017, Oxford University Press.
  2. Lacey, N. 2019). "Python by Example: Learning to Program in 150 Challenges". Cambridge University Press, 2019. ISBN: 9781108716833.
  3. Wesley J. Chun, "Core Python Applications Programming", 3rd Edition , Pearson Education.

- **Reference Books**
  1. Vamsi Kurama, "Python Programming: A Modern Approach", Pearson Education.
  2. Timothy A. Budd, "Exploring Python", Tata MCGraw Hill Education Private Limited 2011, 1 st Edition.
  3. John Zelle, "Python Programming: An Introduction to Computer Science", Second edition, Course Technology Cengage Learning Publications, 2013, ISBN 978- 1590282410
  4. Michel Dawson, "Python Programming for Absolute Beginers" , Third Edition, Course Technology Cengage Learning Publications, 2013, ISBN 978-1435455009

- **Web resources**
  1. https://onlinecourses.swayam2.ac.in/cec22_cs20/preview

Mapping of Cos to POs and PSOs

Course Outcome	**PO Addressed** PO1toPO8	**Correlation Level** L/M/H	**PSO Addressed** PSO1toPSO8	**Correlation Level** L/M/H	**Cognitive Level** K_1toK_6
CO1	PO1,PO4	H	PSO4	H	K_1,K_3
CO2	PO1,PO3,PO4	H	PSO4,PSO5	H	K_4, K_5
CO3	PO4,PO5	H	PSO4	H	K_5, K_6
CO4	PO4	H	PSO4	H	K4,K6
CO5	PO4	H	PSO4	H	K_3

(L–Low, M–Medium ,H –High; K_1 – Remember, K_2 – Understand, K_3– Apply, K_4– Analyze, K_5–Evaluate, K_6 – Create**)**

Course Code: CC2	Python Programming Lab		Credits : 5
Lecture Hours:(L) Per week:	Tutorial Hours: (T)per week	Lab Practice Hours: 5 per week	Total:(L+T+P) Per week: 5
Course Category :Core	Year &Semester: I Year I Semester	Admission Year:	
Pre-requisite	Basic of programming skill		

LearningObjectives:(forteachers:whattheyhavetodointheclass/lab/field)

- Acquire programming skills in core Python.
- Acquire Object-oriented programming skills in Python.
- Develop the skill of designing graphical-user interfaces (GUI) in Python.
- Develop the ability to write database applications in Python.
- Acquire Python programming skills to move into specific branches.

Course Outcomes	Program Outcomes
CO1: Apply basic programming concepts to write Python programs that convert temperature units, calculate student grades, and perform basic arithmetic operations.	PO1,PO2,PO3, PO4,PO5,PO6
CO2: Utilize control flow structures Students will demonstrate proficiency in using if-else statements, loops, and decision-making structures to control program flow.	PO1,PO2,PO3, PO4,PO5,PO6
CO3: Implement data manipulation operations to perform operations on different data structures like lists, tuples, and strings, including accessing, modifying, and iterating over them.	PO1,PO2,PO3, PO4,PO5,PO6
CO4: Develop object-oriented programs to design and implement classes and objects to model real-world entities, demonstrating concepts such as inheritance and polymorphism.	PO1,PO2,PO3, PO4,PO5,PO6
CO5: Design and execute algorithms to implement algorithms to solve mathematical problems, manipulate data, and control program execution flow.	PO1,PO2,PO3, PO4,PO5,PO6

Recap:(notforexamination)Motivation/previouslecture/relevantportionsrequiredforthe course)[Thisisdoneduring2Tutorialhours)

List of Exercises:	Required Hours
1. Write a Python script that prints prime numbers less than 20. 2. Program to calculate total marks, percentage and grade of a student. Marks obtained in each of the five subjects are to be input by user. Assign grades according to the following criteria: Grade A: Percentage >=80 Grade B: Percentage >=70 and 80 Grade C: Percentage >=60 and <70 Grade D: Percentage >=40 and <60 Grade E: Percentage < 40 3. Program, to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user. 4. Program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice. 5. Program to find factorial of the given number using recursive function. 6. Write a Python program to count the number of even and odd numbers from array of N numbers. 7. Write a Python class to reverse a string word by word. 8. Given a tuple and a list as input, write a program to count the occurrences of all items of the list in the tuple. (Input : tuple = ('a', 'a', 'c', 'b', 'd'), list = ['a', 'b'], Output : 3) 9. Create a Savings Account class that behaves just like a Bank Account, but also has an interest rate and a method that increases the balance by the appropriate amount of interest (Hint: use Inheritance). 10. Write a Python program to construct the following pattern, using a nested loop 11. Read a file content and copy only the contents at odd lines into a new file. 12. Create a Turtle graphics window with specific size. 13. Write a Python program for Towers of Hanoi using recursion 14. Create a menu driven Python program with a dictionary for words and their meanings. 15. Devise a Python program to implement the Hangman Game.	75
Total Hours	**75**

Learning Resources:

- **Recommended Texts**

  1. Charles Dierbach, "Introduction to Computer Science using Python - A computational Problem solving Focus", Wiley India Edition, 2015.

  2. Wesley J. Chun, "Core Python Applications Programming", 3rd Edition , Pearson Education, 2016

- **Reference Books**

  1. Mark Lutz, "Learning Python Powerful Object Oriented Programming", O'reilly Media 2018, 5th Edition.
  2. Timothy A. Budd, "Exploring Python", Tata MCGraw Hill Education Private Limited 2011, 1 st Edition.
  3. John Zelle, "Python Programming: An Introduction to Computer Science", Second edition, Course Technology Cengage Learning Publications, 2013, ISBN 978- 1590282410
  4. Michel Dawson, "Python Programming for Absolute Beginers" , Third Edition, Course Technology Cengage Learning Publications, 2013, ISBN 978-1435455009.

Mapping of Cos to POs and PSOs

Course Outcome	**PO Addressed** PO1toPO7	**Correlation Level** L/M/H	**PSO Addressed** PSO1toPSO7	**Correlation Level** L/M/H	**Cognitive Level** K_1toK_6
CO1	PO4	H	PSO4	H	K_3
CO2	PO4	H	PSO4	H	K_3 K_4
CO3	PO4	H	PSO4	H	K_6
CO4	PO4	H	PSO4	H	K_3, K_6
CO5	PO4	H	PSO4	H	K_4 K_5 K_6

(L–Low, M–Medium, H–High;K_1–Remember,K_2–Understand,K_3– Apply, K_4–Analyze, K_5–Evaluate, K_6 – Create**)**

Course Code: SEC-1	Office Automation		Credits: 2
Lecture Hours: (L) per week: 2	Tutorial Hours : (T) per week	Lab Practice Hours: (P)per week	Lecture Hours: (L) per week: 2
Course Category : SEC-1	Year & Semester: I Year I Semester	Admission Year:	
Pre-requisite	Basic skills in Computer operations		

Learning Objectives: (for teachers: what they have to do in the class/lab/field)

- The major objective in introducing the Computer Skills course is to impart training for students in Microsoft Office which has different components like MS Word, MS Excel and Power point.
- The course is highly practice oriented rather than regular class room teaching.
- To acquire knowledge on editor, spread sheet and presentation software.

Course Outcomes	Program Outcomes
CO1:To Understand the basics of computer systems and its components.	PO1,PO2,PO3, PO4,PO5,PO6
CO2: To Understand and apply the basic concepts of a word processing package.	PO1,PO2,PO3, PO4,PO5,PO6
CO3: To Understand and apply the basic concepts of electronic spreadsheet software.	PO1,PO2,PO3, PO4,PO5,PO6
CO4: To Understand and apply the basic concepts of database management system.	PO1,PO2,PO3, PO4,PO5,PO6
CO5: To Understand and create a presentation using PowerPoint tool.	PO1,PO2,PO3, PO4,PO5,PO6

Recap: (not for examination) Motivation/previous lecture/ relevant portions required for the course) [This is done during 2 Tutorial hours)

Units	Contents	Required Hours
I	**Introductory concepts:** Memory unit – CPU-Input Devices: Key board, Mouse and Scanner. Output devices: Monitor, Printer. Introduction to Operating systems & its features: DOS – UNIX– Windows. Introduction to Programming Languages.	6
II	**Word Processing:** Open, Save and close word document; Editing text – tools, formatting, bullets;Spell Checker - Document formatting – Paragraph alignment, indentation, headers and footers, numbering; printing – Preview, options, merge.	6

III	**Spreadsheets:** Excel – opening, entering text and data, formatting, navigating; Formulas – entering, handling and copying; Charts – creating, formatting and printing, analysis tables, preparation of financial statements, introduction to data analytics.	**6**
IV	**Database Concepts:** The concept of data base management system; Data field, records, and files, Sorting and indexing data; Searching records. Designing queries, and reports; Linking of data files; Understanding Programming environment in DBMS; Developing menu drive applications in query language (MS – Access).	**6**
V	**Power point:** Introduction to Power point - Features – Understanding slide typecasting & viewing slides – creating slide shows. Applying special object – including objects & pictures – Slide transition – Animation effects, audio inclusion, timers.	**6**
	Total Hours	**30**

Learning Resources:

- **Recommended Texts**
  1. Peter Norton, "Introduction to Computers" –Tata McGraw-Hill.

- **Reference Books**
  1. Jennifer Ackerman Kettel, Guy Hat-Davis, Curt Simmons, "Microsoft 2003", Tata McGraw-Hill.
- **Web resources :** Web content from NDL / SWAYAM or open source web resources.

Mapping of COs to POs and PSOs

Course Outcome	**PO Addressed** PO1 to PO8	**Correlation Level** L/M/H	**PSO Addressed** PSO1 to PSO8	**Correlation Level** L/ M/ H	**Cognitive Level** K_1 to K_6
CO1	PO1	M	PSO1	M	K_2
CO2	PO1	H	PSO1	H	K_3
CO3	PO1	H	PSO1	H	K_4
CO4	PO1	H	PSO1	H	K_2 K_3
CO5	PO2 PO6	M H	PSO2 PSO3	M H	K_3

(L – Low, M – Medium, H – High; K_1 – Remember, K_2 – Understand, K_3 – Apply, K_4 – Analyze, K_5–Evaluate, K_6 – Create**)**

Course Code: FC1	Problem Solving Techniques		Credits: 2
Lecture Hours: (L) per week: 2	Tutorial Hours : (T) per week	Lab Practice Hours: (P)per week	Lecture Hours: (L) per week: 2
Course Category : FC	Year & Semester :I Year I Semester	Admission Year:	
Pre-requisite	-		

Learning Objectives:

- To understand the importance of algorithms and programs, and to know of the basic problem solving strategies.
- To learn efficient strategies and algorithms to solve standard problems, thus laying a firm foundation for designing algorithmic solutions to problems.

Course Outcomes	Program Outcomes
CO1: To Understand the systematic approach to problem solving.	PO1,PO2,PO3, PO4,PO5,PO6
CO2: The approach and algorithms to solve specific fundamental problems.	PO1,PO2,PO3, PO4,PO5,PO6
CO3: To Understand the efficient approach to solve specific factoring-related problems.	PO1,PO2,PO3, PO4,PO5,PO6
CO4: To Understand the efficient array-related techniques to solve specific problems.	PO1,PO2,PO3, PO4,PO5,PO6
CO5: To Understand the efficient methods to solve specific problems related to text processing.Understand how recursion works.	PO1,PO2,PO3, PO4,PO5,PO6

Recap: (not for examination) Motivation/previous lecture/ relevant portions required for the course) [This is done during 2 Tutorial hours)

Units	Contents	Required Hours
I	**Introduction:** Notion of algorithms and programs – Requirements for solving problems by computer – The problem-solving aspect: Problem definition phase, Getting started on a problem, The use of specific examples, Similarities among	6

	problems, Working backwards from the solution – General problem-solving strategies - Problem solving using top-down design – Implementation of algorithms – The concept of Recursion- Algorithm and Flow Chart.	
II	**Fundamental Algorithms**: Exchanging the values of two variables – Counting - Summation of a set of numbers - Factorial computation - Sine function computation - Fibonacci Series generation - Reversing the digits of an integer – Base Conversion -Flow Chart.	**6**
III	**Factoring Methods**: Finding the square root of a number – The smallest divisor of an integer – Greatest common divisor of two integers - Generating prime numbers – Computing the prime factors of an integer – Generation of pseudo-random numbers - Raising a number to a large power – Computing the *n*th Fibonacci number.	**6**
IV	**Array Techniques**: Array order reversal – Array counting or history programming – Finding the maximum number in a set - Removal of duplicates from an ordered array - Partitioning an array – Finding the k^{th} smallest element – Longest monotone subsequence.	**6**
V	**Text Processing and Pattern Searching**: Text line length adjustment – Left and right justification of text – Keyword searching in text – Text line editing – Linear pattern search -. **Recursive algorithms**: Towers of Hanoi – Permutation generation.	**6**
	Total Hours	**30**

Learning Resources:

- **Recommended Texts**

1. R. G. Dromey, *How to Solve it by Computer*, Pearson India, 2007.

- **Reference Books**

1. George Polya, Jeremy Kilpatrick, *The Stanford Mathematics Problem Book: With Hints and Solutions*, Dover Publications, 2009 (Kindle Edition 2013).

Greg W. Scragg, *Problem Solving with Computers*, Jones & Bartlett 1st edition, 1996.

Mapping of COs to POs and PSOs

Course Outcome	**PO Addressed** PO1 to PO7	**Correlation Level** L/M/H	**PSO Addressed** PSO1 to PSO7	**Correlation Level** L/ M/ H	**Cognitive Level** K_1 to K_6
CO1	PO1	H	PSO1	H	K_2
CO2	PO1,PO4	M	PSO1,PSO4	M	K_2
CO3	PO4	H	PSO4	H	K_3 K_4
CO4	PO4	M	PSO4	M	K_3K_4
CO5	PO4	H	PSO4	M	K_4 K_5

(L – Low, M – Medium, H – High; K_1 – Remember, K_2 – Understand, K_3 – Apply, K_4 – Analyze, K_5–Evaluate, K_6 – Create**)**

First Year (Semester – II)

Course Code: CC3	**Data Structures & Algorithms**		**Credits: 5**
Lecture Hours: (L) per week: 4	**Tutorial Hours : (T) per week**	**Lab Practice Hours: (P)per week**	**Total: (L+T+P) per week: 4**
Course Category : CC3	**Year & Semester: I Year II Semester**	**Admission Year:**	
Pre-requisite	Basic knowledge in data and representations		
Links to other Courses			

Learning Objectives: (for teachers: what they have to do in the class/lab/field)

- To impart the basic concepts of data structures and algorithms.
- To acquaint the student with the basics of the various data structures and make the students knowledgeable in the area of data structures.
- This course also gives insight into the various algorithm design techniques

Course Outcomes	**Program Outcomes**
CO1: To introduce the concepts of Data structures and to understand simple linear data structures.	PO1,PO2,PO3, PO4,PO5,PO6
CO2: Learn the basics of stack data structure, its implementation and application.	PO1,PO2,PO3, PO4,PO5,PO6
CO3: Use the appropriate data structure in context of solution of given problem and demonstrate a familiarity with major data structures.	PO1,PO2,PO3, PO4,PO5,PO6
CO4: To introduce the basic concepts of algorithms.	PO1,PO2,PO3, PO4,PO5,PO6
CO5: To give clear idea on algorithmic design paradigms like Dynamic Programming, Backtracking, Branch and Bound.	PO1,PO2,PO3, PO4,PO5,PO6

Recap: (not for examination) Motivation/previous lecture/ relevant portions required for the course) [This is done during 2 Tutorial hours)

Units	**Contents**	**Required Hours**
I	**INTRODUCTION TO DATA STRUCTURES:** • **Data Structures:** Definition- Time & Space Complexity, • **Arrays:** Representation of arrays, Applications of arrays, sparse matrix and its representation, • **Linear list:** Singly linked list implementation, insertion, deletion and searching operations on linear list • **Circular linked list:** implementation, Double linked list implementation,	**15**

	insertion, deletion and searching operations. Applications of linked lists- Dynamic Storage management.	
II	**STACKS & QUEUES:** • Operations, array and linked representations of stack, • Stack applications, infix to postfix conversion, postfix expression evaluation, recursion implementation • **Queues**: operations on queues, array and linked representations. • **Circular Queue:** operations,, applications of queues.	**15**
III	**TREES & GRAPHS:** • **Trees:** Definitions and Concepts- Representation of binary tree, Binary tree traversals (**In order, Post order , preorder**), • **Binary search trees**. • **Graphs** : Representation of Graphs- Types of graphs -Breadth first traversal – Depth first traversal- -Applications of graphs.	**15**
IV	**INTRODUCTION TO ALGORITHMS:** • **INTRODUCTION**: Definition of Algorithms- Overview and importance of algorithms- pseudocode conventions, Asymptotic notations, practical complexities. • **Divide-and-Conquer:** Binary Search- Quick Sort- Merge Sort. • **Greedy Method:** Knapsack problem- Tree vertex splitting.	**15**
V	**DYNAMIC PROGRAMMING, BACKTRACKING & BRANCH & BOUND** • **Dynamic programming**: General method, Multistage Graphs. • **Backtracking:** ,Subset Problem ,8 Queens, Graph coloring. • **Branch & Bound**: Hamiltonian Circle, Travelling salesperson problem.	**15**
	Total Hours	**75**

Learning Resources:

- **Recommended Texts**

1. Ellis Horowitz , Sartaj Sahni, Susan Anderson Freed, Second Edition , "Fundamentals of Data in C", Universities Press

2. E. Horowitz, S. Sahni and S. Rajasekaran, Second Edition ,"Fundamentals of Computer Algorithms " Universities Press

- **Reference Books**

1. Seymour Lipschutz ,"Data Structures with C", First Edition, Schaum's outline series in computers, Tata McGraw Hill.

2. R.Krishnamoorthy and G.Indirani Kumaravel, Data Structures using C, Tata McGrawHill – 2008.

3. A.K.Sharma, Data Structures using C , Pearson Education India,2011.

4. G. Brassard and P. Bratley, “Fundamentals of Algorithms”, PHI, New Delhi, 1997.
5. 4, . A.V. Aho, J.E. Hopcroft, J.D. Ullmann,, “The design and analysis of Computer
6. Algorithms”, Addison Wesley, Boston, 1974

- **Web resources:** Web resources from NDL Library, E-content from open source libraries

Mapping of COs to POs and PSOs

Course Outcome	**PO Addressed** PO1 to PO7	**Correlation Level** L/M/H	**PSO Addressed** PSO1 to PSO7	**Correlation Level** L/ M/ H	**Cognitive Level** K_1 to K_6
CO1	PO1	H	PSO6	H	K_2
CO2	PO1	H	PSO6	M	K_3
CO3	PO1	H	PSO1	H	K_3
CO4	PO4	M	PSO4	H	K_3
CO5	PO4	H	PSO4	H	K_3 K_4

(L – Low, M – Medium, H – High; K_1 – Remember, K_2 – Understand, K_3 – Apply, K_4 – Analyze, K_5–Evaluate, K_6 – Create**)**

Course Code: CC3	Data Structures & Algorithms Lab		Credits: 5
Lecture Hours:(L) Per week	**Tutorial Hours: (T)per week**	**Lab Practice Hours: (P)per week: 5**	**Lecture Hours:(L) Per week**
Course Category: CC3	**Year& Semester: I Year II Semester**	**Admission Year:**	
Pre-requisite	Basic skills in problem solving		

LearningObjectives:(forteachers:whattheyhavetodointheclass/lab/field)

- To understand and implement basic data structures using C
- To apply linear and non-linear data structures in problem solving.
- To learn to implement functions and recursive functions by means of data structures
- To implement searching and sorting algorithms

Course Outcomes	Program Outcomes
CO1: To Implement data structures using C	PO1,PO2,PO3, PO4,PO5,PO6
CO2: To Implement various types of linked lists and their applications	PO1,PO2,PO3, PO4,PO5,PO6
CO3: To Implement Tree Traversals	PO1,PO2,PO3, PO4,PO5,PO6
CO4: To Implement various algorithms in C	PO1,PO2,PO3, PO4,PO5,PO6
CO5: To Implement different sorting and searching algorithms	PO1,PO2,PO3, PO4,PO5,PO6

Recap:(not for examination)Motivation/previous lecture/relevant portions required for the course)[This is done during 2 Tutorial hours)

	List of Exercises:	Required Hours
	Implement the following exercises using C Programming language: 1. Array implementation of stacks 2. Array implementation of Queues 3. To implement Array using binary Search. 4. To implement Array using Quick Sort. 5. To implement Array using Merge Sort. 6. Binary Tree traversal using (in order – preorder - post order) 7. Implementation Of DFS And BFS of Graph. 8. Finding all pairs of shortest path of Graph. 9. Finding single source Shortest path of graph. 10. To implantation of Infix to postfix conversion.	**75**
	Total Hours	**75**

Learning Resources:

- **Recommended Texts**

  1. Ellis Horowitz , Sartaj Sahni, Susan Anderson Freed, Second Edition , "Fundamentals of Data in C", Universities Press

  2. E. Horowitz, S. Sahni and S. Rajasekaran, Second Edition ,"Fundamentals of Computer Algorithms " Universities Press

- **Reference Books**

  1. Seymour Lipschutz ,"Data Structures with C", First Edition, Schaum's outline series in computers, Tata McGraw Hill.
  2. .2. R.Krishnamoorthy and G.Indirani Kumaravel, Data Structures using C, Tata McGrawHill – 2008.
  3. A.K.Sharma, Data Structures using C , Pearson Education India,2011.
  4. . G. Brassard and P. Bratley, "Fundamentals of Algorithms", PHI, New Delhi, 1997.
  5. 4, . A.V. Aho, J.E. Hopcroft, J.D. Ullmann,, "The design and analysis of Computer
  6. Algorithms", Addison Wesley, Boston, 1974
  7. 5. Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, Introduction to Algorithms, Third edition, MIT Press, 2009
  8. Sanjoy Dasgupta, C.Papadimitriou and U.Vazirani , Algorithms , Tata McGraw-Hill, 2008.

- **Web resources:** Web resources from NDL Library, E-content from open source libraries

Mapping of Cos to POs and PSOs

Course Outcome	**PO Addressed** PO1toPO7	**Correlation Level** L/M/H	**PSO Addressed** PSO1toPSO7	**Correlation Level** L/M/H	**Cognitive Level** K_1toK_6
CO1	PO4	H	PSO4	H	K_2 K_3
CO2	PO4	H	PSO4	H	K_3
CO3	PO4	H	PSO4	H	K_3
CO4	PO4	H	PSO4	H	K_3K_4
CO5	PO4	H	PSO4	H	K_3 K_4

(L–Low,M–Medium,H–High;K_1–Remember,K_2–Understand,K_3– Apply, K_4–Analyze, K_5–Evaluate, K_6 – Create**)**

Course Code: SEC-2	Quantitative Aptitude		Credits: 2
Lecture Hours: (L) per week: 2	Tutorial Hours : (T) per week	Lab Practice Hours: (P)per week	Lecture Hours: (L) per week: 2
Course Category :SEC-2	Year & Semester :I Year II Semester	Admission Year:	
Pre-requisite	Basic knowledge in numerical ability		

Learning Objectives: (for teachers: what they have to do in the class/lab/field)

- To improve the quantitative skills of the students
- To prepare the students for various competitive exams

Course Outcomes	Program Outcomes
CO1: Apply mathematical concepts to apply in HCF, LCM, decimal fractions, simplification, square roots, and cube roots to solve numerical problems efficiently.	PO1,PO2,PO3, PO4,PO5,PO6
CO2: Problem-solving skills in various topics including averages, problems on numbers, ages, percentage, profits and loss, ratio and proportion, partnership, and chain rule.	PO1,PO2,PO3, PO4,PO5,PO6
CO3: Understanding of time-related problems to solve problems related to time and work, pipes and cisterns, time and distance, problems on trains, boats and streams, simple interest, compound interest, logarithms, area, volume, surface area, aces, and games of skill.	PO1,PO2,PO3, PO4,PO5,PO6
CO4: Application of permutation, combination, and probability to apply permutation, combination, and probability concepts to solve problems related to these topics.	PO1,PO2,PO3, PO4,PO5,PO6
CO5: Data interpretation skills in interpreting data represented in various forms such as calendars, clocks, stocks and shares, data representation, tabulation, bar graphs, pie charts, and line graphs.	PO1,PO2,PO3, PO4,PO5,PO6

Recap: (not for examination) Motivation/previous lecture/ relevant portions required for the course) [This is done during 2 Tutorial hours)

Units	Contents	Required Hours
I	Numbers - HCF and LCM of numbers - Decimal fractions - Simplification - Square roots and cube roots - Average - problems on Numbers.	6
II	Problems on Ages - Surds and Indices - percentage - profits and loss - ratio and proportion - partnership - Chain rule.	6
III	Time and work - pipes and cisterns - Time and Distance - problems on trains - Boats and streams - simple interest - compound interest -	6

	Logarithms - Area - Volume and surface area - races and Games of skill.	
IV	Permutation and combination - probability - True Discount - Bankers Discount-Height and Distances - Odd man out & Series.	**6**
V	Calendar - Clocks - stocks and shares - Data representation - Tabulation - Bar Graphs - Pie charts - Line graphs.	**6**
	Total Hours	**30**

Learning Resources:

- **Recommended Texts**

1. . "Quantitative Aptitude", R.S. AGGARWAL., S. Chand & Company Ltd.,

- **Web resources: Authentic** Web resources related to Competitive examinations

Mapping of COs to POs and PSOs

Course Outcome	**PO Addressed** PO1 to PO8	**Correlation Level** L/M/H	**PSO Addressed** PSO1 to PSO8	**Correlation Level** L/ M/ H	**Cognitive Level** K_1 to K_6
CO1	PO1	H	PSO1	H	K_3
CO2	PO3	H	PSO3	M	K_4, K_5
CO3	PO1	H	PSO1	H	K_3, K_4
CO4	PO1	M	PSO4	M	K_3, K_4
CO5	PO3	H	PSO3	H	K_4, K_5

(L – Low, M – Medium, H – High; K_1 – Remember, K_2 – Understand, K_3 – Apply, K_4 – Analyze, K_5–Evaluate, K_6 – Create**)**

Course Code: SEC-3	Advanced Excel		Credits: 2
Lecture Hours: (L) per week: 2	Tutorial Hours : (T) per week	Lab Practice Hours: (P)per week	Lecture Hours: (L) per week: 2
Course Category : SEC-3	Year & Semester :I Year II Semester	Admission Year:	
Pre-requisite	Basic knowledge in office automation / Excel		

Learning Objectives: (for teachers: what they have to do in the class/lab/field)

The objective of this course is to help the students learn the advanced features of Excel, to summaries, analyze, explore, and present visualizations of data in the form of charts, graphs.

Course Outcomes	Program Outcomes
CO1: To Observer the Basic knowledge of Microsoft Excel. Advanced Functionality Mastery to utilize advanced Excel functions effectively, such as writing conditional expressions, using logical functions, and implementing lookup and reference functions like VLOOKUP with various match types and nested structures.	PO1,PO2,PO3, PO4,PO5,PO6
CO2: To Aggregate numeric data and summarize into categories and subcategories.	PO1,PO2,PO3, PO4,PO5,PO6
CO3: Data Validation and Template Design to set up data validations for specifying valid ranges and lists, create templates for standardizing worksheets, and design the structure of templates for efficient data entry and management.	PO1,PO2,PO3, PO4,PO5,PO6
CO4: To Create pivot tables to consolidate data from multiple files.	PO1,PO2,PO3, PO4,PO5,PO6
CO5: Advanced Formatting and What-If Analysis to learn advanced formatting techniques, including auto and conditional.	PO1,PO2,PO3, PO4,PO5,PO6

Recap: (not for examination) Motivation/previous lecture/ relevant portions required for the course) [This is done during 2 Tutorial hours)

Units	Contents	Required Hours
I	Basics of Excel- Customizing common options- Absolute and relative cells- Protecting and un-protecting worksheets and cells- Working with Functions - Writing conditional expressions - logical functions - lookup and reference functions- V look UP with Exact Match, Approximate Match- Nested V look UP with Exact Match- V look UP with Tables, Dynamic Ranges- Nested V look UP with Exact Match- Using V Look UP to consolidate Data from Multiple Sheets	6

II	Data Validations - Specifying a valid range of values - Specifying a list of valid values- Specifying custom validations based on formula - Working with Templates Designing the structure of a template- templates for standardization of worksheets - Sorting and Filtering Data -Sorting tables- multiple-level sorting- custom sorting- Filtering data for selected view - advanced filter options- Working with Reports Creating subtotals- Multiple-level subtotal.	**6**
III	Creating Pivot tables Formatting and customizing Pivot tables- advanced options of Pivot tables- Pivot charts- Consolidating data from multiple sheets and files using Pivot tables- external data sources- data consolidation feature to consolidate data- Show Value As % of Row, % of Column, Running Total, Compare with Specific Field- Viewing Subtotal under Pivot- Creating Slicers.	**6**
IV	More Functions Date and time functions- Text functions- Database functions- Power Functions - Formatting Using auto formatting option for worksheets- Using conditional formatting option for rows, columns and cells- What If Analysis - Goal Seek- Data Tables- Scenario Manager.	**6**
V	Charts - Formatting Charts- 3D Graphs- Bar and Line Chart together- Secondary Axis in Graphs- Sharing Charts with PowerPoint / MS Word, Dynamically- New Features Of Excel Sparklines, Inline Charts, data Charts- Overview of all the new features.	**6**
	Total Hours	**30**

Learning Resources:

- **Recommended Tex**

  Excel 2019 All-in-One For Dummies – 2018- Greg Harvey

- **Reference Books**

  Microsoft Excel 2019 Pivot Table Data Crunching-2019,Bill Jelen and Michael Alexander

- **Web resources:** Web resources from NDL Library, E-content from open source libraries.

Mapping of COs to POs and PSOs

Course Outcome	**PO Addressed** PO1 to PO7	**Correlation Level** L/M/H	**PSO Addressed** PSO1 to PSO7	**Correlation Level** L/ M/ H	**Cognitive Level** K_1 to K_6
CO1	PO1	H	PSO1	H	K_3
CO2	PO1	H	PSO1	M	K_3 K_4
CO3	PO5	L	PSO5	L	K_3 K_5
CO4	PO5	M	PSO5	M	K_3K_4
CO5	PO5	M	PS05	M	K_3 K_5

(L – Low, M – Medium, H – High; K_1 – Remember, K_2 – Understand, K_3 – Apply, K_4 – Analyze, K_5–Evaluate, K_6 – Create**)**