SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total Credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28501	**DISCRETE STRUCTURES**	3	0	0	3	-	70	30	-	-	3

COURSE OBJECTIVES:
Throughout the course, students will be expected to demonstrate their understanding of Discrete Mathematics by being able to do each of the following:

1. Gain knowledge to solve problems by use of Mathematical Induction.
2. Construct graph and tree for Real world problem.
3. Use counter examples and apply logical reasoning, Recurrence relation to solve a variety of problems.

COURSE OUTCOMES: After completion of the course, students will be able to:

CO1 Explain various types of set, relations, functions and algebraic structures.
CO2 Apply the logic theory to find the validity of an argument.
CO3 Apply Mathematical Induction and counting techniques to solve different problems.
CO4 Apply the graph theory techniques to solve real world problems.
CO5 Describe the concepts of tree, binary search tree, spanning trees using different algorithms.
CO6 Solve the recurrence relations using various techniques.

THEORY:
COURSE CONTENTS:

Unit1: Introduction, sets, finite and infinite sets, uncountable infinite sets, mathematical induction, principles of inclusion and exclusion and multi sets. Relations and Functions: A relational model for data bases, properties of binary relations, equivalence relations and partitions, partial ordering relations and lattices, chains and anti-chains, job scheduling problem, functions and pigeonhole principle.

Unit2: Logic-Propositional calculus, conjunction, disjunctions and negation, validity and consistency, computability and formal language: Russell's paradox and non-computability, ordered sets, languages, phrase structure grammars, types of grammars and languages.

Unit3: Graphs: Introduction and basic terminology of graphs, weighted graphs and multigraphs,
introduction to paths and circuits, shortest path in weighted graph: Dijkstra algorithm, warshall's algorithm, and pruning algorithm.

Unit4: Euler's path, Hamiltonian paths and circuits, traveling salesperson problem, planar graphs,. Introduction to trees, rooted trees, path length in rooted trees, prefix codes,

introduction to binary search tree, spanning trees and cut-sets, minimal spanning tree, Kruskal's algorithm, prim's algorithm.

Unit5: Introduction to recurrence relations and recursive algorithms: Linear recurrence relations with constant coefficients, homogeneous solutions, particular solutions, total solutions, solutions by the method of generating functions. Introduction to groups rings integral domains and fields.

TEXT BOOKS RECOMMENDED:

1) Lieu C.L, "Elements of Discrete Mathematics", Pearson Education.
2) Schaum's Series, "Discrete Mathematics", 2nd Edition, TMH Publication.
3) Kenneth H. Rosen, "Discrete Mathematics and its Application", 5th edition, TMH Publication

REFERENCE BOOKS:

1) Kolman, "Discrete Mathematical Structures", 5th Edition, Pearson Education.

ASSESSMENT TOOLS :

Direct assessment: End-Sem Examination, Mid-Term Test, Class Assignments**,** Quiz**,** Attendance

Indirect assessment: Course End Survey

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	3	2	2	2	1	-	-	-	-	-	-	1	1	1	-
CO2	3	3	2	2	2	-	-	-	-	-	-	1	1	1	-
CO3	3	3	2	2	2	-	-	-	-	-	-	1	2	1	-
CO4	3	3	2	2	2	-	-	-	-	-	-	1	2	1	-
CO5	3	3	2	2	2	-	-	-	-	-	-	1	2	1	-
CO6	3	3	2	2	1	-	-	-	-	-	-	1	1	1	-

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total Credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28506	COMPUTER ORGANIZATION & ARCHITECTURE	3	0	2	3	1	70	30	40	60	4

PRE-REQUISITES: Digital Logic Design.

COURSE OBJECTIVES:

1. Conceptualize the basics of organizational and architectural issues of a digital computer and Classify and compute the performance of machines.
2. Learn about various data transfer techniques in digital computer and the I/O interfaces.
3. Estimate the performance of various classes of Memories.
4. Explain the basics of hardwired and micro-programmed control of the CPU, pipelined architectures and Hazards.

COURSE OUTCOMES: After completion of course, students will be able to:

CO1 Describe basic components and design of a computer, including CPU, memories, and input/output units.
CO2 Apply different memory mapping techniques to improve the performance of CPU.
CO3 Explain the instruction cycle the I/O related concepts.
CO4 Explain the general concepts in digital logic design, including logic elements, and their use in combinational and sequential logic circuit design.
CO5 Apply the pipelining concepts to implement the parallel processing.
CO6 Explain the basic concept of assembly language.

THEORY:
COURSE CONTENTS:

Unit 1: Basics of Computer, Generation of Computers, Classification of Computers, Basic operational concepts, Von Neumann Architecture, Bus structures, Hardware and Software Interfaces. General register organization, Memory and Computer registers, Stack organization, Instruction formats, Addressing modes, Data transfer and manipulation, Register Transfer Language, Register transfer Bus and Memory transfer, Micro-operations.

Unit 2: Instruction codes, computer instructions, Instruction Execution, Timing and Control, Instruction cycle, Memory-Reference Instructions, and I/O and Interrupt cycle, Machine Language, Assembly Language, Assembler, Program loops, Programming Arithmetic and logic operations, subroutines

Unit 3: Input output interface, Asynchronous data transfer, Modes of transfer, Priority interrupt, Input-output and interrupt Direct Memory access(DMA), Control unit, Design of Control unit, Control Memory, Address sequencing, Binary Arithmetic, Add, Subtract, Multiply Divide Algorithms.

Unit 4: Memory Hierarchy, Memory Classification, Modern Memory Technologies. Associative Memory, Cache Memory: Cache Memory Principles, Elements of Cache Design. Mapping Functions: Direct Mapping, Associative Mapping, Set-Associative Mapping. Virtual Memory, Address Space and Memory Space, Address Mapping Using Pages, Page Replacement Algorithm.

Unit 5: Introduction to Pipelining, Basic Performance Issue in Pipelining, The Major Hurdle of Pipelining, Pipelining Hazards: Structural Hazard, Data Hazard, Control Hazard. Concepts of Risc And Cisc, Instruction Execution Characteristics, Reduced Instruction Set Architecture, Risc Pipelining, Risc v/s Cisc Controversy. Parallel Organization: Multiprocessing, Cache Coherence, Vector Computation, Parallel Processors.

TEXT BOOKS RECOMMENDED:

1. William Stallings, "Computer Organization and Architecture", 7th Edition, Pearson Education.
2. C. V. Carl Hamacher, Zvonks Vranesic, Safea Zaky "Computer Organization", 5th Edition, McGraw Hill.

REFERENCE BOOKS:

1. Morris Mano, "Computer System Architecture",5th Edition, Prentice Hall of India.
2. S. Andrew, S. Tanenbaum, "Structured Computer Organization", 4th Edition, PHI/Pearson.
3. John P. Hayes , "Computer Architecture and Organization", 3rd Ed. McGraw Hill.

ASSESSMENT TOOLS :

Direct assessment: End-Sem Examination, Mid-Term Test, Class Assignments**,** Quiz**,** Attendance

Indirect assessment: Course End Survey

PRACTICAL:

LAB ASSIGNMENTS:

1. Introduction to assembly language programming.
2. Print the string in assembly language programming.
3. Assembly language program to implement for getting the input from user.
4. Implement the arithmetic instruction in assembly language prograaming.
5. Implement the logical instruction in assembly language programming.
6. Implement the assembly program for find the even and odd number.
7. Implement an assembly program to find the largest number.
8. Implement the assembly program to find the factorial of a number.

ASSESSMENT TOOLS:
Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test
Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	2	2	1	-	-	-	-	-	-	-	-	-	-	-	-
CO2	3	2	1	-	-	-	-	-	-	-	-	-	-	-	-
CO3	2	2	1	1	-	-	-	-	-	-	-	-	-	1	1
CO4	2	2	1	1	-	-	-	-	-	-	-	-	-	1	1
CO5	3	2	1	-	1	-	-	-	-	-	-	-	-	1	1
CO6	2	1	1	-	-	-	-	-	-	-	-	-	-	1	1

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total Credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28503	SOFTWARE ENGINEERING	3	0	2	3	1	70	30	40	60	4

PRE-REQUISITES: None

COURSE OBJECTIVES:
1. To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.
2. To provide an idea of using various process models in the software industry according to given circumstances.

COURSE OUTCOMES: After completion of course, students will be able to:
CO1 Explain various process used in SDLC and project management.
CO2 Analyze all fact related to measurement of development of software. Estimate time, cost, efforts, team size, size of software etc.
CO3 Determine appropriate modelling approach for software development.
CO4 Design software as per the requirement of the end users.
CO5 Apply standard software testing principles along with writing manual test cases.
CO6 Describe Software Quality Assurance and its approaches.

THEORY:
COURSE CONTENTS:

UNIT 1: Software characteristics, components, applications. The software process, various software process models, SDLC. Project management concepts, software process and project metrics: software measurements. COCOMO model. Risk management, Scheduling: time charts, tracking the schedule.

UNIT 2: System engineering, Information engineering, Product engineering systems analysis. Requirements analysis, analysis principles, Analysis specification. Elements of Analysis modelling: data modelling, behavioural modelling. Creating an ERD, DFD, and control flow Model, data dictionary.

UNIT 3: Design principles, Design concepts: functional independence, cohesion, and coupling. Design documentation, principles of design specification. Transform Mapping, Transaction Mapping, Software Modelling and UML Project planning.

UNIT 4: Software testing fundamentals: objectives, principles, testability, test case design. White box testing: basis path testing. Black box testing, Testing documentation. Software

verification and validation: unit testing, integration testing, regression testing, alpha and beta testing. Case studies, Study of Testing tools (QTP, Rational Robot, Winrunner, Loadrunner)

UNIT 5: Software quality assurance: software quality, SQA activities, formal approaches to SQA, software reliability. Software configuration management: SCM process. Metrics for analysis, design and source code testing. Introduction to formal methods, software reuse, software re-engineering, CASE tools

TEXT BOOKS RECOMMENDED:

1. Pressman, R. S, "Software Engineering – A Practitioner's Approach", 7th Edition, McGraw Hill.
2. Ian Sommerville, "Software Engineering",7th Edition, Pearson Education.
3. Waman S Jawadekar, "S/w Engg Principles & Practice",9th Edition, Tata McGraw Hill.

REFERENCE BOOKS:

1. PankajJalote, "An integrated Approach to software Engineering", 3rd Edition, Narosa Pub House.
2. GezziC.Mandrioli D. Jazayeri M., "Fundamentals of Software Engineering", 2nd Edition, Pearson Education 1996.
3. Dr. K.V. K. Prasad "Software Testing Tools" by Dreamtech Press .Richard H.Thayer, "Software Engineering & Project Managements", Willey India.

ASSESSMENT TOOLS:
Direct assessment: End-Sem Examination, Mid-Term Test, Class Assignments**,** Quiz**,** Attendance
Indirect assessment: Course End Survey

PRACTICAL:

List of Experiments:

1. Prepare a **Software Requirement Specification (SRS)** document for real life problem.
2. To perform the function oriented diagram: Data Flow Diagram (DFD)
3. To perform the user's view analysis for the suggested system: Use case diagram.
4. To design the structural view diagram for the system: Class diagram, object diagram.
5. To design the behavioral view diagram : State-chart diagram, Activity diagram
6. To perform the behavioral view diagram for the suggested system: Sequence diagram, Collaboration diagram.
7. To perform the implementation view diagram: Component diagram for the system.
8. To perform the environmental view diagram: Deployment diagram for the system.
9. To perform various testing using the testing tool unit testing, integration testing for a sample code of the suggested system.
10. Perform Estimation of effort using FP Estimation for chosen system.
11. To prepare time line chart/Gantt Chart/PERT Chart for selected software project.

ASSESSMENT TOOLS:
Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test
Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	3	2	1	-	-	1	-	-	-	-	1	1	2	-	1
CO2	3	3	1	2	2	-	-	-	2	-	-	1	1	-	1
CO3	3	3	2	2	1	1	-	-	2	-	-	1	1	2	1
CO4	3	2	3	1	1	-	-	-	2	-	1	1	2	2	1
CO5	3	2	2	1	2	-	-	-	1	-	-	1	1	1	-
CO6	3	1	2	1	-	1	-	-	-	-	-	1	1	-	-

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total Credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28841	**SYSTEM PROGRAMMING**		2	2		3			40	60	3

PRE-REQUISITES: Fundamentals knowledge of programming and C language.

COURSE OBJECTIVES:
This course is intended to
1. Provide an understanding of the basic concepts of System programming
2. Familiarize the student with different language processors.
3. Provide an ability to design different system programs.

COURSE OUTCOMES: After completion of the course, students will be able to:
CO1 Describe the basics of system programming
CO2 Explain the role of different language processors.
CO3 Explain the process of scanning and parsing.
CO4 Illustrate the sequence of steps in compilation, linking and loading.
CO5 Explain the working of program development tools like editors and debuggers.
CO6 Describe the basics of operating system goals and functions.

COURSE CONTENTS:
THEORY:

UNIT 1: Introduction: Need of System Program, Distinction with Application Programming, Language Processing Activities, Fundamentals of Language Processing, Fundamental of Language Specifications, Languages Processor Development Tools, Review of 'C' Language, Data Structures for Language Processing.

UNIT 2: Assemblers and Macro Processors: Overview of Assembly Language Program, Elements of Assembly Language Programming, A Simple Assembly Scheme, Pass Structure of Assemblers, Design of a Two Pass Assembler. Macro Definition and Macro Call, Macro Expansion, Nested Macro Calls, Advanced Macro Facilities, Design of A Macro Processor.

UNIT 3: Scanning and Parsing: Parsing, Pars Tree and Abstract Syntax Tree, Top Down Parsing, Bottom Up Parsing, LEX, YACC. Compilers and Interpreters: Aspects of Compilation, Phases of Compilation, Memory Allocation, Compilation of Expression, Compilation of Control Structure, Code Optimization. Compiling Using GCC, Interpreters: Benefits of Interpreter, Overview of Interpretation, JVM.

UNIT 4: Linkers and Loader: Reallocation and Linking Concepts, Linking Methods : Static Linking, Dynamic Linking, Library Linking, Object Linking and Embedding (OLE), Module Linking. Design of Linker, Self Reallocating Program, Linking for Overlays, Linkers v/s Loaders, Loader: Types of Loaders, Sequential and Direct Loader, Compiler and Go Loader, General Loading Scheme.

UNIT 5: Operating Systems: Fundamental Principles of OS Operations, OS Interaction with Computer and User Programs, Structure of Operating Systems. Classes of Operating Systems, Virtual Machine Operating Systems and Modern Operating Systems. Editors and Debuggers: Overview of Editing Process, User Interface, Editor Structure, Debugging Functions and Capabilities, Relationship With Other Parts of The System, User Interface Criteria. Debugging Using GDB.

TEXT BOOKS RECOMMENDED:

1. D.M. Dhamdhere, "System Programming", 2011, Tata Mcgraw Hill.
2. L.L. Beck ,"An Introduction to System Programming", 3rd Edition ,Pearson Education.
3. Srimanta Pal ,"System Programming", 2nd Edition, Oxford University Press.

REFERENCE BOOKS:

1 J.J. Donovan "System Programming ", 2nd Edition Second, Tata Mcgraw Hill.

2 J.D. Ullmann, Alfred V. Aho, R. Sethi ,"Principles of Compiler Design", 2nd Edition, Pearson Education.

PRACTICAL:

List of Experiments:

1. Implementation of basic c program.
2. Implementation of basic file handling program.
3. Implementation of basic file handling program with menu driven concept.
4. Implementation of a line text editor.
5. Applications of different types of Macros.
6. Implementation of scanning and parsing.
7. study of lex and yacc tools.
8. Debugging Using GDB.

ASSESSMENT TOOLS:

Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test

Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	2	1	-	-	-	-	-	-	-	-	-	1	2	1	2
CO2	2	3	1	2	2	-	-	-	-	-	-	1	-	1	1
CO3	1	3	2	2	-	-	-	-	-	-	-	1	1	1	1
CO4	3	2	1	-	-	-	-	-	-	-	2	2	1	1	2
CO5	2	1	2	-	-	-	-	-	-	-	-	2	2	2	1
CO6	3	1	1	-	2	2	-	-	-	-	-	1	2	1	2

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
DEPARTMENT OF INFORMATION TECHNOLOGY

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total Credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28998	**System Software Laboratory-II**	**-**	**-**	**2**	**-**	**1**	**-**	**-**	**40**	**60**	**1**

PRE-REQUISITES: Data Structure

COURSE OBJECTIVE:
1. To acquire programming skills in core Python.
2. To acquire Object Oriented Skills in Python.
3. To develop the ability to write database applications in Python

COURSE OUTCOMES: After the completion of course, student will be able to:
CO1 Use data structures and string in python programming.
CO2 Write basic programs of python language using control flow statement.
CO3 Describe the basic Package of python.
CO4 Write functions in python.
CO5 Implement python program for file handling and exception tracing.
CO6 Solve real world problems using advanced python libraries.

THEORY:
COURSE CONTENTS:

UNIT 1: Introduction: History of Python, Need of Python Programming, Applications. Basics of Python Programming, Variables, Assignment, Keywords, Input-Output, Indentation. Types - Integers, Strings, Booleans.

UNIT 2: Operators and Expressions: Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations.

UNIT 3: Data Structures: Lists - Operations: Traversing a List, list operations, list slices, list methods, Slicing, Methods; Tuples: Tuple assignment, tuple as return value, Sets, Dictionaries, Sequences.

UNIT 4: Control Flow Statements: The if Decision Control Flow Statement, The if…else Decision Control Flow Statement, The if…elif…else Decision Control Statement, Nested if Statement, The while Loop, The for Loop, The continue and break Statements, Catching Exceptions Using try and except Statement, Functions, Built-In Functions, Commonly Used Modules, Function Definition and Calling the Function, The return Statement and void Function, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Scope and Lifetime of Variables in a function - Global and Local Variables.

UNIT 5: FILES, MODULES, PACKAGES: Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, Brief introduction of these packages: PANDAS, NUMPY, SCIKIT-LEARN.

TEXT BOOKS RECOMMENDED:

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist, 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016
2. Guido van Rossum and Fred L. Drake Jr, ―An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.

REFERENCE BOOKS:

1. Core Python Programming, W.Chun, Pearson.
2. Introduction to Python, Kenneth A. Lambert, Cengage

PRACTICAL:
LAB ASSIGNMENTS:
Experiment No.1

1. Write a program to demonstrate basic data type in python.
2. Write a program for checking if the given number is even or odd.
3. Write a program that takes 2 numbers as command line arguments and prints its sum.
4. Write a program that prompts the user to input the length and the width of a rectangle and outputs the area and circumference of the rectangle.
5. Write a program to read a person's age from the keyboard and display whether he is eligible for voting or not.

Experiment No.2

1. Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
2. Write a program to check if the given year is a leap year or not.
3. Write a program to find the sum and average of given numbers using lists.
4. Write a program to find the minimum and maximum elements in the lists.
5. Write a program to use split and join methods in the string and trace a birthday of a person with dictionary data structure.
6. Write a program which accepts a sequence of comma-separated numbers from the console and generates a list and a tuple which contains every number. Suppose the following input is supplied to the program: 34, 67, 55, 33, 12, 98. Then, the output should be: ['34', '67', '55', '33', '12', '98'] ('34',67', '55', '33', '12', '98').
7. With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line.

Experiment No.3

1.Write a Python program to create the multiplication table (from 1 to 10) of a number.
Expected Output:
Input a number: 6
6 x 1 = 6
6 x 2 = 12

6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60

2. Write a Python program to check if a triangle is equilateral, isosceles or scalene.
Note :An equilateral triangle is a triangle in which all three sides are equal.
A scalene triangle is a triangle that has three unequal sides.
An isosceles triangle is a triangle with (at least) two equal sides.

3. Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700 (both included) Write a Python program to convert temperatures to and from celsius,fahrenheit.[Formula : c/5 = f-32/9 [where c = temperature in celsius and f = temperature in fahrenheit]Expected Output : 60°C is 140 in Fahrenheit 45°F is 7 in Celsius

4.Write a Python program to guess a number between 1 to 9. Note : User is prompted to enter a guess. If the user guesses wrong then the prompt appears again until the guess is correct, on successful guess, user will get a "Well guessed!" message, and the program will exit.

5. Write a Python program to count the number of even and odd numbers from a series of numbers.

6. Write a Python program to construct the following pattern, using a nested for loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

7.Write a Python program to check the validity of password input by users.
Validation :
At least 1 letter between [a-z] and 1 letter between [A-Z].
At least 1 number between [0-9].
At least 1 character from [$#@].
Minimum length 6 characters.
Maximum length 16 characters.

8. Write a Python program to check whether an alphabet is a vowel or consonant.
Expected Output: Input a letter of the alphabet: k
k is a consonant.

9.Write a Python program to convert a month's name to a number of days.

Experiment No.4

1.Create an inner function to calculate the addition in the following way

- Create an outer function that will accept two parameters, a and b
- Create an inner function inside an outer function that will calculate the addition of a and b

- At last, an outer function will add 5 into addition and return it.

2. Assign a different name to function and call it through the new name.

3.Write a Python function to multiply all the numbers in a list.
Sample List : (8, 2, 3, -1, 7)
Expected Output : -336

4.Write a Python function to check whether a number is perfect or not.
According to Wikipedia : In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).
Example : The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.
5. Write a Python program to access a function inside a function.

Experiment No.5

1. Write a code to accept two numbers and display the quotient. An appropriate exception should be raised if the user enters the second number (Denominator) as zero.
2. Write a code to raise an exception in which if the user purchases an item from a shop and his credit card limit does not fulfill the amount criteria of purchased things.
3. Writing an exception handler for Exception to catch all the exceptions.
4. Write a code where you use the wrong number of arguments for a method (sqrt( ) and Pow( )).Use the exception handling process to catch the value Error exception.
5. How to handle exceptions If Local or Global variable name is not defined in file.

Experiment No. 6

Consider a Iris dataset which contains the following information:

- The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.
- Attribute Information:

1. sepal length in cm

2. sepal width in cm

3. petal length in cm

4. petal width in cm

5. class:

-- Iris Setosa

-- Iris Versicolour

-- Iris Virginica

Write a Python program to do the following tasks for the above Iris dataset:

- Reading the dataset “Iris.csv”.

- Displaying up the top rows of the dataset with their columns.
- Displaying the number of rows randomly.
- Displaying the number of columns and names of the columns.
- Displaying the shape of the dataset.
- Display the whole dataset.
- Slicing the rows of the dataset.
- Displaying only specific columns.
- Calculating sum, mean and mode of a particular column.
- Cleaning and detecting missing values.
- Plotting graph For IRIS Dataset Using Seaborn Library And matplotlib.pyplot library

ASSESSMENT TOOLS:

Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test

Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO1 0**	**PO1 1**	**PO1 2**	**PSO 1**	**PSO 2**	**PSO 3**
CO 1	2	2	2	2	3	1	-	-	-	-	-	1	-	2	1
CO 2	2	3	3	2	3	1	-	-	-	-	-	1	-	2	1
CO 3	3	3	4	3	3	2	-	-	-	-	-	1	-	2	1
CO 4	3	3	3	2	3	1	-	-	-	-	-	1	-	2	1
CO 5	3	3	3	3	3	1	-	-	-	-	-	1	1	2	3
CO 6	3	3	3	3	3	1	-	1	2	1	-	1	1	3	3

SHRI G.S. INSTITUTE OF TECHNOLOGY & SCIENCE, INDORE
INFORMATION TECHNOLOGY DEPARTMENT

Subject Code	Subject Name	L	T	P	Th. Credit	Pr. Credit	Maximum Marks				Total credit
							End Sem	Class Work	Sessional Work	End Sem	
IT28999	SCIENTIFIC COMPUTING	-	-	2	-	1	-	-	60	40	1

PER-REQUISITES: Basic Knowledge of Programming.

COURSE OBJECTIVES:
This course is intended to
1. Build an understanding of the fundamental concepts of R programming
2. Familiarize the student with programming in R
3. Allow the student to gain expertise in sampling data.

COURSE OUTCOMES: After completing this course student will able to:
CO1 Create basic programs using vectors in R.
CO2 Implement Arrays , List and Matrices in R.
CO3 Apply loops and Functions in R.
CO4 Apply file input and outputs.
CO5 Use Packages in R.
CO6 Apply sampling of data.

THEORY:
COURSE CONTENTS:

UNIT 1: Overview of R, Installation of R, R studio, R Basics, Vectors, sequences and Integers, logical vectors, character vectors.

UNIT 2: Objects & Attributes, Arrays, Matrices List, Data Frame.

UNIT 3: Loops & Functions: for() loop, the apply() function,the if() statement,the while() loop, the repeat{} and break statement and Functions.

UNIT 4: File Input Outputs,Plots, Low level Graphic Functions and High Level Graphic Functions.

UNIT 5: Sampling: Univariate Distribution, Multivariate Distribution,R object oriented system [S3],[S4]and R[5], Packages.

TEXT BOOKS RECOMMENDED:

1. Kahate Paul Gerrard ,Radia M. Johnson, “Mastering Scientific Computing with R” PACKT publishing.
2. Sandip Rakshit, “R Programming for Beginners” McGraw-Hill
3. Hadley Wickhem , “R for Data Science”, O Reilly

ASSESSMENT TOOLS:
Direct assessment: Lab Assignments, Quiz, Viva-Voce examination (Internal and External), Attendance, Written Test
Indirect assessment: Course End Survey, External Examinar Feedback

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES (H-3, M-2, L-1, or '-'):

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	1	-	-	-	-	-	-	-	-	-	2	-	-	-
CO2	2	2	-	-	-	-	-	-	-	-	-	2	-	-	-
CO3	2	2	-	-	-	-	-	-	-	-	-	2	-	-	-
CO4	2	3	-	-	-	-	-	-	-	-	-	2	-	-	-
CO5	2	3	2	-	-	-	-	-	-	-	-	2	-	-	-
CO6	2	2	2	2	-	-	-	-	-	-	-	3	-	2	2
Avg	2	2	1	-	-	-	-	-	-	-	-	2	-	-	-