



# MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

## COURSE CONTENT

ESSENTIALS OF PROBLEM-SOLVING USING PYTHON LAB								
II Semester: CE / CSD / CSE / CSM / ECE / EEE / ME								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
2420572	Foundation	L	T	P	C	CIA	SEE	Total
		0	0	2	1	40	60	100
Contact Classes: NIL	Tutorial Classes: NIL	Practical Classes: 30			Total Classes: 30			
Prerequisites: A course on "Problem solving using C and C++"								

### Course Overview:

This course gives acquaintance to Python Programming and Graph Theory. The course deals with Python programming concepts and concepts in graph theory like properties of standard graphs, Eulerian graphs, Hamiltonian graphs, Chordal graphs, Distances in graphs, Planar graphs, graph connectivity and Colouring of graphs. Graph theory is used in Network Topologies and Routing Algorithms, Algorithm Design, Social Network Design, Logistics.

### Course Objectives:

1. Basic building blocks of python.
2. Using of Functions and Modules.
3. Importance of Multithreading in problem solving.
4. The fundamental concepts of graph theory.
5. Graph coloring and traversal algorithms for solving real-world problems.

### Course Outcomes: After Completion of the Course, Students should be able to

1. Apply fundamental Python programming concepts including data types, operators, built-in functions, control flow statements, sequences, dictionaries, and sets to develop structured and efficient code for solving basic computational problems.
2. Write modular Python programs using user defined and built in functions, functional programming concepts, and modules/packages to promote code reuse, organization, and maintainability.
3. Develop Python applications that perform file handling and input/output operations, and implement multithreaded programs using Python's threading modules while understanding concepts like processes and the Global Interpreter Lock (GIL).
4. Analyze and apply fundamental graph theory concepts including various types of graphs, graph representations, isomorphisms, and classical problems such as Eulerian circuits and Hamiltonian cycles to model and solve real world problems.
5. Implement and evaluate graph algorithms including shortest path algorithms, graph coloring, traversal techniques, and minimum spanning tree algorithms to solve

## **LIST OF EXPERIMENTS:**

### **Exercise 1: Python Numbers**

- a. You are developing a program to determine whether a given year is a leap year, using the following formula: a leap year is one that is divisible by four, but not by one hundred, unless it is also divisible by four hundred. For example, 1992, 1996, and 2000 are leap years, but 1967 and 1900 are not. The next leap year falling on a century is 2400.
- b. You are developing a program to determine the greatest common divisor and least common multiple of a pair of integers.
- c. You are developing a program to create a calculator application. Write code that will take

### **Exercise 2: Control Flow**

- a) Write a Program for checking whether the given number is a prime number or not.
- b) Write a program to print Fibonacci series up to given n value.
- c) Write a program to calculate factorial of given integer number.
- d) Write a program to calculate value of the following series  $1+x-x^2+ -x.t..x$ .
- e) Write a program to print Pascal triangle.

### **Exercise 3: Python Sequences**

- a) Write a program to sort the numbers in ascending order and strings in reverse alphabetical order.
- b) Given an integer value, return a string with the equivalent English text of each digit. For example, an input of 89 results in "eight-nine" being returned. Write a program to implement it.
- c) Write a program to create a function that will return another string similar to the input string, but with its case inverted. For example, input of "Mr. Ed" will result in "mR.eD" as the output string.
- d) Write a program to take a string and append a backward copy of that string, making a palindrome.

### **Exercise 4: Python Dictionaries**

- a) Write a program to create a dictionary and display its keys alphabetically.
- b) Write a program to take a dictionary as input and return one as output, but the values are now the keys and vice versa.
- c) Given a List, extract all elements whose frequency is greater than K. Ex: Input test list= (4,6,4,3,3,4,3,4,3,8], k=3 Output - [4,3]

### **Exercise 5: Files**

- a) write program to compare two text files. If they are different, give the line and column numbers in the files where the first difference occurs.
- b) write a program to compute the number of characters, words and lines in a file.

### **Exercise 6: Functions**

- a. Write a function ball collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding. Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius If (distance

between two balls centers)  $\leq$  (sum of their radii) then (they are colliding)

- b. Find mean, median, mode for the given set of numbers in a list.
- c. Write simple functions `max2()` and `min2()` that take two items and return the larger and smaller item, respectively. They should work on arbitrary Python objects. For example, `max2(4, 8)` and `min2(4, 8)` would each return 8 and 4, respectively.
- d. Write a function `nearly equal` to test whether two strings are nearly equal. Two strings `a` and `b` are nearly equal when `a` can be generated by a single mutation on `b`.
- e. Write a function `dups` to find all duplicates in the list.
- f. Write a function `unique` to find all the unique elements of a list.
- g. Write a function `cumulative_product` to compute cumulative product of a list of numbers.
- h. Write a function `reverse` to reverse a list. Without using the reverse function
- i. Write function to compute GCD, LCM of two numbers. Each function shouldn't exceed one line.

### Exercise 7: Files

- a) Write a program to compare two text files. If they are different, give the line and column numbers in the files where the first difference occurs.
- b) Write a program to compute the number of characters, words and lines in a file.

### Exercise 8: Multithreading

- a. Write a program to create thread using `thread` module.
- b. Write a program to create thread using `threading` module.
- c. Write a Program to use Python's `threading` module to calculate the square and cube of a number concurrently.

### Exercise 9:

- a. Write a Python program to implement Euler Circuit.
- b. Write a Python program to implement Dijkstra's algorithm.
- c. Given a connected graph  $G$  with  $N$  nodes and  $M$  edges (edges are bi-directional). Every node is assigned a value  $A[i]$ . We define a value of a simple path as: Value of path = Maximum of (absolute difference between values of adjacent nodes in a path). A path consists of a sequence of nodes starting with start node  $S$  and end node  $E$ .  $S-u_1-u_2-\dots-E$  is a simple path if all nodes on the path are distinct and  $S, u_1, u_2, \dots, E$  are nodes in  $G$ . Given start node  $S$  and end node  $E$ , find the minimum possible "value of path" which starts with node  $S$  and ends with node  $E$ .
- d. Yatin created an interesting problem for his college juniors. Can you solve it? Given  $N$  rooms, where each room has a one-way door to a room denoted by  $room[i] \ (1 \leq i \leq N)$  a positive integer  $K$  such that, if a person starts from room 1 and continuously moves to the room it is connected to (i.e.  $room[i]$ ), the person should end up in room  $i$  after  $K$  Steps, Note: The condition should hold for each room. If there are multiple possible values.

### Exercise 10: Implement the following using python

- a.  $M$ -coloring
- b. Vertex coloring
- c. Edge coloring

**Exercise 11:** Implement the following graph traversal methods.

a. Depth-First Search

b. Breadth-First Search

c. 1 You are presented with a network comprising  $N$  computers and  $M$  wired connections between them. Your objective is to precisely  $K$  wires from your inventory. computers that can be linked together determine and report the size of the establishing these connections. optimize the network's connectivity using The aim is to maximize the number of within the given constraints. Your task is to largest network that can be formed by In the context of this problem, computers are considered connected if they share either a direct or indirect wired connection. It is worth noting that the value of  $K$  will always be less than the number of isolated (standalone) networks in the given configuration, and it may even be zero.

d. A country consists of  $N$  cities. These cities are connected with each other using  $N-1$  bidirectional roads that are in the form of a tree. Each city is numbered from 1 to  $N$ . You want to safeguard all the roads in the country from any danger, and therefore, you decide to place cameras in certain cities. A camera in a city can safeguard all the roads directly connected to it. Your task is to determine the minimum number of cameras that are required to safeguard the entire country.

**Exercise 12:** Travelling Salesman problem.

a) You are working in a salesmen company as a programmer. There are  $n$  towns in your country and  $m$  directed roads between them. Each road has a cost person should spend on fuel. The company wants to sell goods in all  $n$  towns. There are infinitely many salesmen in the company. We can choose some positive number of salesmen and give a non-empty list of towns to each of them. Towns from the list are the towns to sell goods in. Each salesman will visit all the towns in his list in this particular order in cycle (after the last town he will return to the first town and so on). Salesman can visit other towns on his way but he will not sell goods in these towns. Two Salesmen cannot sell goods in one town because it will attract unnecessary attention to your company. But for every town there must be a salesman who sell goods in this town. spend money on fuel. If salesman's list of towns consists of exactly one town then he should pay fee to stay in this town each month (each town has its own fee) or he should go for a round trip and cycle. Your task is to calculate the minimal amount of money company must spend monthly to achieve its goals. We will assume that every salesman will spend a month to make one

b) It is the final leg of the most famous amazing race. The top ' $n$ ' competitors have made it to the final. The final race has just begun. The race has ' $m$ ' checkpoints. Each team can reach any of the ' $m$ ' checkpoint but after a team reaches a particular checkpoint that checkpoint gets closed and is not open to any other team. The race ends when ' $k$ ' teams finish the race. Each team travel at a constant speed throughout the race which end the race. might be different for different teams. Given the coordinates of  $n$  teams and  $m$  checkpoints and speed of individual team return the value of minimum time needed to

c) Little Jhool is a very lenient teaching assistant in his college. He doesn't like cutting the marks of students, so obviously, every student in his tutorial loves him. But anyway, the teacher has got to know about the leniency of Jhool while giving marks, so this time in exam, he decides to give a different exam paper to every single student to check how well have the students been taught by Jhool. Now, Little Jhool knows the strong and weak topics of every single student, so he wants to maximize the total marks obtained by students in his tutorial .You are given the number of

students in Jhool's tutorial, denoted by  $n$  -  $n$  also being the number of different exam papers - that is, one for every student. Every student will get only one exam paper to solve. You are further given a matrix,  $(n \times n)$  denoting the marks every student will get if he attempts a particular exam obtained by his entire class.

**Exercise 13:** Construct minimal spanning tree using the following

a) Prim's Algorithm

b) Kruskal's Algorithm

c) There are total  $N$  Hacker-cities in plane. Each city on coordinates  $(X^*t, Y^*E)$  and there can be any number of cities on the same coordinates. is located You have to make these cities connected by constructing some roads in such a way that it is possible to travel between every pair of cities by traversing the roads. The cost of constructing one road between any two cities is the minimum of the absolute difference between their  $X$  and  $Y$  coordinates. has you want to earn more and more. you decided to do this in the most optimal way possible, such that the total cost of constructing these roads is minimal. You have to return the minimum money you need to spend on connecting all the cities.

### TEXT BOOKS:

1. Core Python Programming, Wesley J. Chun, Third Edition, Pearson.
2. Karin R Saoub, Graph Theory: An Introduction to Proofs, Algorithms, and Applications, 1<sup>st</sup> edition, Chapman and Hall, 2021.

### REFERENCE BOOKS:

1. Think Python, Allen Downey, Green Tea Press
2. Introduction to Python, Kenneth A. Lambert, Cengage
3. Python Programming: A Modern Approach, VamsiKurama, Pearson
4. Learning Python, Mark Lutz, O'Reilly

### ELECTRONIC RESOURCES:

1. <https://www.geeksforgeeks.org/python-programming-language/>
2. <https://www.w3schools.com/python/>
3. <https://www.programiz.com/python-programming>
4. <https://docs.python.org/3/tutorial/>
5. <https://www.codechef.com/practice>

### MATERIALS ONLINE:

1. Course template
2. Open-ended experiments