

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -2 EXAMINATION- 2024

B.Tech-3rd Semester (CSE/IT/BT/BI)

COURSE CODE (CREDITS): L-18B11CI314 (3)

MAX. MARKS: 25

COURSE NAME: Python Programming Essentials

COURSE INSTRUCTORS: Dr. Monika, Dr. Naveen, Dr. Ramesh, Dr. Aman, Mr. Kuntal

MAX. TIME: 1 Hour 30 Minutes

Note: (a) All questions are compulsory.

(b) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems

Q. No.	Question	CO	Marks
Q1.	<p>What will be the value of ans after executing the below code ?</p> <pre>class myrange(): def __iter__(self): self.start = 'a' self.end = 'e' return self def __next__(self): if self.start < self.end: x = self.start self.start = chr(ord(self.start) + 1) else: raise StopIteration return x _range = iter(myrange()) ans = "" for i in _range: ans += i</pre>	CO-3 & CO-4	4
	<p>What is the output?</p> <pre>class A(object): def __init__(self,x=10): self._x = x class B(A): def __init__(self): A.__init__(self, 20) def display(self): print(self._x) def main(): obj = B() obj.display() main()</pre>		
Q2.	<p>What is the output?</p> <pre>import re string = '\n and \r are escape sequences.' result = re.findall(r'[\n\r]', string) print(result)</pre>	CO-3	5
	<p>If the input is "ankitrai326" and "ankirai@", what will be the output of the following code?</p> <pre>import re regex = '[a-zA-z0-9]\${' def check(string): if(re.search(regex, string)): print("Accept") else: print("Discard")</pre>		

	<ul style="list-style-type: none"> • Reads content from a text file named input.txt. • Counts the number of words in the file and prints the word count. • Handle exceptions if the file is not found. 		
Q3.	<p>a) What is operator overloading in Python?</p> <ul style="list-style-type: none"> • Explain the concept of operator overloading in object-oriented programming. • Give an example where Python allows overloading of an operator using special methods. <p>b) Case Study: Design and implement a Python program to demonstrate inheritance, polymorphism, and operator overloading.</p> <ul style="list-style-type: none"> • Create a base class Employee with attributes name and salary. • Implement two subclasses: Manager and Developer. • Override the <code>__str__()</code> method to provide customized string representation for each type of employee. • Overload the <code>+</code> operator to combine the salaries of two Employee objects. • Demonstrate polymorphism by printing the string representation and total salary using operator overloading. 	CO-4	5
Q4.	<p>a) What is the purpose of regular expressions in Python?</p> <ul style="list-style-type: none"> • Explain the role of regular expressions in string manipulation. • Provide an example of a regular expression that matches a valid email address. <p>b) Case Study: Design and implement a Python program that performs the following using string methods and regular expressions:</p> <ul style="list-style-type: none"> • Validate a list of email addresses and print valid ones. Extract phone numbers from a given text (phone numbers can have the format +91-1234567890 or 123-456-7890). • Count and print the frequency of each word in a given text. 	CO-3	5
Q5.	Write a function that takes a list of tuples and returns a list of tuples with duplicates removed, while maintaining the original order.	CO-1	3
Q6.	Explain the role of assertions in Python. How can assertions help in debugging?	CO-2	3