Uni. RN													

UNITED COLLEGE OF ENGINEERING & RESEARCH, PRAYAGRAJ (010)		Department of Computer Science & Engineering	
First Sessional Examination (ODD Semester 2022-23)		SEMESTER:**III**	Date:-12/10/2022
TIME: **2 hours.**	SUBJECT:**Python Programming Solution**	Paper code:KNC302	MM. **30**

READ ALL INSTRUCTIONS AND QUESTIONS VERY CAREFULLY

		SECTION A (Attempt ALL questions) Very short answer	**[5]**	**CO**	**Bloom's Taxonomy Level**
1	a	Explain the advantages and disadvantages of using python. Solution: **Advantages** / **Disadvantages** It is easy to learn and use, and it has an extensive library. / Because of its elementary programming, users face difficulty while working with other programming languages. Python increases productivity. / Python is a time-consuming language. It has a low execution speed. It is very flexible. / There are many issues with the design of the language, which only gets displayed during runtime. It has a very supportive community. / It is not suited for memory-intensive programs and mobile applications.	[1]	1	Understand (L2)

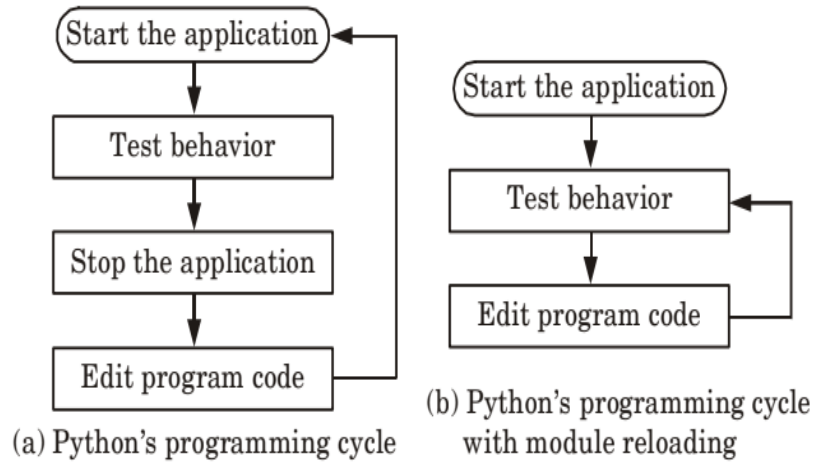
1	b	Explain local variables and global variables in Python.
		Solution:
		Local variables in python are those variables that are declared inside the function. Alternatively, they are said to defined within a local scope. A user can only access a local variable inside the function but never outside it.
		Global Variables
		Variables that are created outside of a function are known as global variables.
		Global variables can be used by everyone, both inside of functions and outside.
		The global Keyword
		Normally, when you create a variable inside a function, that variable is local, and can only be used inside that function.
		To create a global variable inside a function, you can use the global keyword.
		`def myfunc():`
		`  global x`
		`  x = "fantastic"`
		
		`myfunc()`
		
		`print("Python is " + x)`
			[1]	1	Understand (L2)

1	c	What will be the output of the following python code? **for i in range (10):** **If (i = = 5):** **break** **print (i)** Solution: 0 1 2 3	[1]	2	Apply (L3)

Uni. RN													

		4			

1	d	Explain Nested loop with example.
		Solution:
		for i in range(5):
		for j in range(5):
		print(i,j)
			[1]	2	Understand (L2)

1	e	Implement the Python program using function to swap two numbers without using third variable. Solution: def swap_num(a,b): a,b=b,a print("a ",a) print("b ",b)	[1]	3	Apply (L3)
		SECTION B (Attempt any two questions) Long answer	**[10]**		
2		Explain Programming Cycle of Python in detail with a suitable diagram. Solution:  	[5]	1	Remember (L1)
3		Explain Arithmetic Operators, Assignment Operators, Comparison Operators, Logical Operators and Bitwise Operators with a suitable python program. Solution: Python Arithmetic Operators Operator Name Example + Addition x + y - Subtraction x - y * Multiplication x * y / Division x / y % Modulus x % y ** Exponentiation x ** y // Floor division x // y Python Assignment Operators Operator Example = x = 5 x = 5 += x += 3 x = x + 3 -= x -= 3 x = x - 3 *= x *= 3 x = x * 3	[5]	1	Understand (L2)

Uni. RN													

		/= x /= 3 x = x / 3 %= x %= 3 x = x % 3 //= x //= 3 x = x // 3 **= x **= 3 x = x ** 3 &= x &= 3 x = x & 3 |= x |= 3 x = x | 3 ^= x ^= 3 x = x ^ 3 >>= x >>= 3x = x >> 3 <<= x <<= 3x = x << 3 Python Comparison Operators Operator Name Example == Equal x == y != Not equal x != y > Greater than x > y < Less than x < y >= Greater than or equal to x >= y <= Less than or equal to x <= y Python Logical Operators Operator Description Example and Returns True if both statements are true x < 5 and x < 10 or Returns True if one of the statements is true x < 5 or x < 4 not Reverse the result, returns False if the result is true not(x < 5 and x < 10) Python Bitwise Operators Operator Name Description & AND Sets each bit to 1 if both bits are 1 | OR Sets each bit to 1 if one of two bits is 1 ^ XOR Sets each bit to 1 if only one of two bits is 1 ~ NOT Inverts all the bits << Zero fill left shift Shift left by pushing zeros in from the right and let the leftmost bits fall off >> Signed right shift Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off			
4		Explain Python IDE and write the name of any four python IDE? Explain applications of python in detail. 1. IDE is a software package that consists of several tools for developing and testing the software. 2. An IDE helps the developer by automating the process. 3. IDEs integrate many tools that are designed for SDLC. 4. IDEs were introduced to diminish the coding and typing errors. 5. Some of the Python IDEs are pycharm,pydev,spyder,idle. Python Applications 1) Web Applications We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing.It also provides Frameworks such as Django, Pyramid, Flask etc to design web based applications 2) Desktop GUI Applications Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wxWidgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.	[5]	1	Remember (L1)

Uni. RN													

		3) Software Development Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc. 4) Scientific and Numeric Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics. 5) Business Applications Python is used to build Bussiness applications like ERP and e-commerce systems. Tryton is a high level application platform. 6) Console Based Application We can use Python to develop console based applications. For example: IPython. 7) Audio or Video based Applications Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc. 8) 3D CAD Applications To create CAD application Fandango is a real application which provides full features of CAD. 9) Enterprise Applications Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc. 10) Applications for Images Using Python several application can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc.			
SECTION C (Attempt any two question) Long answer			**[10]**		

5		Implement the python program to construct the following pattern using nested for loop:
		`        *`
		`      * * *`
		`    * * * * *`
		`  * * * * * * *`
		`* * * * * * * * *`
		
		`rows = int(input("Enter number of rows: "))`
		`for i in range(1, rows + 1):`
		`  for k in range(1, (rows - i) + 1):`
		`  print(end="  ")`
		`  for j in range(2*i-1):`
		`  print("* ",end='')`
		`  print()`
			[5]	2	Apply (L3)

6		Implement the program in python to count the number of vowels, consonants and number in a given string.
		Solution:
		`str1=input("Enter any string")`
		`str1=str1.lower()`
		`v=c=n=0`
		`for i in str1:`
		`  if(i=='a' or i=='i' or i=='e' or i=='o' or i=='u'):`
			[5]	2	Apply (L3)

Uni. RN													

		v=v+1
		elif(i.isdigit()):
		n=n+1
		elif(i==' '):
		continue
		else:
		c=c+1
		print("Vowel",v,"Numbers ",n,"Consonants ",c)			

7		Explain loops in python. Implement a programusing loop to display elements from a given list present at odd index positions. Solution: l=[1,2,3,4,5,6,6,7,8] for i in range(1,len(l),2): print(l[i])	[5]	2	Apply (L3)
		SECTION D (Attempt any one question) Long answer	**[5]**		

8		Implement the program in python to reverse a string without using python function.
		Solution:
		
		def reverse(s):
		str = ""
		for i in s:
		str = i + str
		return str
		
		or
		
		def reverse(string):
		string = string[::-1]
		return string
			[5]	3	Apply (L3)

9		Implementthe program in python to display all prime numbers from 1 to 100 using function.
		Solution:
		
		def checkPrime(num):
		
		# 0, 1 and negative numbers are not prime
		if num < 2:
		return 0
		else:
		x = num // 2
		for j in range(2, x + 1):
		if num % j == 0:
		return 0
		
		# the number would be prime if we reach here
		return 1
		
		a, b = 1, 100
		for i in range(a, b + 1):
			[5]	3	Apply (L3)

Uni. RN													

		if checkPrime(i): print(i, end=" ")			

Course Outcome Wise Marks Distribution	CO1	CO2	CO3	CO4	CO5
	12	12	6	-	-

Bloom's Taxonomy Wise Marks Distribution	L1	L2	L3	L4	L5	L6
	10	8	27	-	-	-