USN										

CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 – March 2019

Sub:	Python Application Programming					Sub Code:	15CS664	Branch:	CSE
Date:	07/03/2019	Duration:	90 min's	Max Marks:	50	Sem / Sec:	6th A/B / C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1 (a)	Explain the following: i)Skills necessary for a programmer ii) Interactive mode iii)Short Circuit evaluation of expression iv) Modulus Operator i) Knowledge about Programming Language • Vocabulary and Grammar • i.e. Awareness about syntax and right standard of programming Problem solving skill • How to frame solutions for a given problem ii) Python has two basic modes: normal and interactive. The normal mode is the mode where the scripted and finished .py files are run in the Python interpreter. Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. iii) Evaluate the following Logical Expression for x=1 and x=8 having y=2: x>=2 and (x/y)>2. When x=1, the first condition fails and the 'and' is observed. So is no need for the interpreter to check the other condition x/y>2. Even if condition 2 evaluates to be true the entire statement is going to return false. Hence python stops executing the second condition after finding first to be false which is called as "Short Circuiting" iv) Works on integers and yields remainder when the first operand is divided by the second E.g.: 12%10 will be evaluated to 2	[04]	CO1	L2
(b)	Describe Python language support for arithmetic operators. Write a Python program to calculate and print the student total marks based on 2 exam, one sport event and 1 activity conducted in a college with a weightage of an activity=20% , each exam=30% and sports=20% for 100 marks. **Arithmetic Operators :** + : Addition, -: Subtraction, * : Multiplication, / : division ** : Exponentiation, % : Modulus • Division operator in Python 2.x will truncate result to int. • Division operator in Python 3.x will not truncate the results i.e. output will be in type float **Program :** m1,m2=int(input("Enter the marks scored by a student in 2 exams out of 100")).split() s1=int(input("Enter the marks scored by a student in sports out of 100")) a1= int(input("Enter the marks scored by a student in activity out of 100")) a1=(a1/100)*20	[06]	CO1	L3

	s1=(s1/100)*20 m1=(m1/100)*30 m2=(m2/100)*30 total=s1+a1+m1+m2 print(“total marks scored by student is %d out of 100” %(total))			

2 (a)	List and give syntax of all Python supported conditional statements along with its usage with an example program to check whether a given number is positive or negative or zero.
	
	1 Mark per each syntax and 1 mark for the program
	• If
	• If else
	• Chained conditionals : If elifelse
	• Nested Conditionals : nested if else
	If statement
	Syntax :
	if (condition):
	statements
	Program :
	if(a==0):
	print(“zero")
	if(a>0):
	print(“+VE")
	if(a<0):
	print(“-VE")
	
	If else statement
	Syntax :
	if (condition):
	statements
	else :
	statements
	Program :
	if(a==0):
	print(“zero")
	else :
	if(a>0):
	print(“+VE ")
	else:
	print(“-VE ")
	If elif else statement : Chained Conditionals
	Syntax :
	if (condition):
	statements
	elif(condition) :
	statements
	else:
	statements
	Program :
	if(a==0):
	print(“zero")
		[06]	CO1	L2

	elif(a<0):
	  print("Negative")
	else:
	  print("Positive")
	Nested Conditionals
	if (condition):
	  statements
	else :
	 if (condition):
	  statements
	 else :
	  statements
	Program :
	if (a==0):
	 print(“zero”)
	else :
	 if (a<0):
	  print(“Negative”)
	 else :
	  print(“Positive”)			

(b)	Explain the rules of precedence used by Python to evaluate an expression. • When more than one operator is present in an expression then they are evaluated in the order of precedence - PEMDAS. 1.Parentheses : 2*(3-1) is 4 2.Exponentiation : 2**1+1 is 3 and not 4 3.Multiplication & Division have same precedence 4.Addition and subtraction have same precedence • Operators with same precedence are evaluated from left to right.	[04]	CO1	L2
3 (a)	How Python handles the exceptions? Explain with an example program. • Exception is an error that happens during execution of a program. • Try Block : Include the sequence of instructions which may have a problem while execution • Except Block: Sequence of statements that want to be executed when an error is encountered. • Finally Block: Executes always regardless of result of try and except blocks • If exception doesn’t occurs then statements in the except block is skipped. • If exception occurs then control transfers from try block to the except block. **Program:** try : inp=input(“enter farenheitvalue”) fahr=float(inp) cel=(fahr-32.0)*5.0/9.0 print(cel) except: print(“Enter correct value pls”)	[06]	CO1	L2

(b)	Predict the output and justify your answer: i) -11%9 ii)7.7//7 iii)(200-70)*10/5 iv)not "False" • 7 • 1.0 • 260.0 • False	[04]	CO1	L3

P.T.O.

P.T.O

4 (a)	List and explain any four built in string manipulation functions supported in Python with examples. • upper() - returns the upper case value a='hello' print(a.upper()) #prints 'HELLO' • lower() returns the lower case value a='hELllo' print(a.lower()) #prints 'hello' • strip( ) - removes the white space (spaces, tabs or newlines) from the beginning and end of a string and not in the middle a=' hello' print(a.strip( )) # prints 'hello' as output • find('element_to_be_searched') searches for the position of one string within another a='hello' print(a.find('l')) #prints 2 as output	[05]	CO2	L2
(b)	Write a user defined function "roll_dice( )" which returns random numbers between 1 to 6. **Program:** def roll_dice(): print(random.randint()) roll_dice()	[05]	CO1	L2

5 (a)	A positive integer 'm' is a sum of squares if it can be written as k+x, where k>0 and x>0 and both k and x are perfect squares. Write a Python function "Sum_of_squares(m)" that takes an integer 'm' and returns true if 'm' is a sum of squares and False otherwise. [Hint:Sum_of_squares(41) should return True, Sum_of_squares(30) should return False, Sum_of_squares(17) should return true]
	Program :
	import math
	def sumofsquares(m):
	r=math.floor(math.sqrt(m))
	for j in range(1,r+1):
		[10]	CO1	L3

	for k in range(r,0,-1):
	if j**2 + k** ==m:
	return True
	return False			

6 (a)	List the rules to declare a variable in Python. Demonstrate at least three different types of variable uses with an example program. • Can start with any letter "A to Z" or "a to z" or an Underscore followed by Zero or letters, Underscore and digits • Cannot start with digits • Python does not allow punctuation characters such as @,&,$,% as an identifier • Python is a case Sensitive language • Variable uses i.e. Label, Access **Program :** A=10 B=12.0 C= "PYTHON" D=A*B print("A:%d,B:%f,C:%f,D:%s" %(A,B,C,D))	[05]	CO1	L1

(b)	Write a Python code to print the following pattern using loops:
	# x x
	x # x
	x x #
	Program :
	for i in range(3):
	for j in range(3):
	if i==j:
	print("#")
	print(x)
		[05]	CO2	L2

7	Write a Python program which repeatedly reads numbers until the user enters "done". Once done is entered print out the total, count and average of the numbers. If the user enters anything other than the numbers, detect their mistake using try and except and print an error message and skip to the next number.
	Program :
	count = 0
	total = 0
	avg = 0
	keepgoing = True
	while keepgoing:
	prompt1 = 'Enter a number \n'
	line = input (prompt1)
	try:
	line = float(line)
		[10]	CO2	L2

```
    count = count + 1
    total = total + line
    avg = total / count
  except:
   if line == 'Done' or line == 'done':
     break
   else:
     print 'Invalid Input'
     continue
print count, total, avg
```