USN

Internal Assessment Test 1 –
Nov 2024

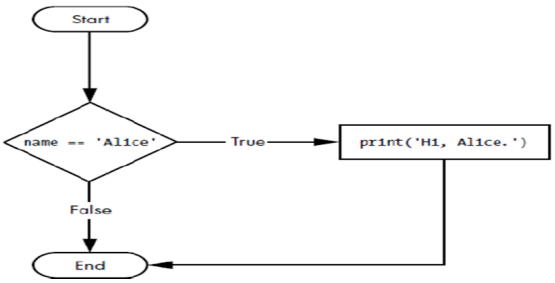
Sub:	Introduction to Python Programming					Sub Code:	BPLCK105B	Branch:	Chemistry Cycle
Date:	22-11-2024	Duration:	90 min's	Max Marks:	50	Sem/Sec:	I / Chemistry Cycle		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1.a	Explain the usage of following functions with example code i) input() ii) print() iii) range() iv) len() v)type() **i) input():** The input() function waits for the user to type some text on the keyboard and press ENTER eg: name=input("Enter your name:") output: "Enter your name: RAM **ii) print():** print() function is used to print the expression written in double quotes (" ") eg: print("Hello World") output: "Hello World" **iii)range():** The Python range() function returns a sequence of numbers, in a given range. eg: for i in range(5): print(i) output: 0 1 2 3 4 **iv)len():** we pass string in len() function and this function evaluates integer value of number of characters in that string eg: len("Hello") output: 5 **v)type():** when type(obj) is passed, it returns the type of the given object. eg: print(type(5)) output: <class int>	[5]	CO1	L2
1.b	Explain about Control statements (if, else, elif) with example •An **if statement's** clause (that is, the block following the if statement) will execute if the statement's condition is True. The clause is skipped if the condition is False.	[5]	CO1	L2

➢ Example:

```
if name == 'Alice':
    print('Hi, Alice.')
```

➢ Flowchart:



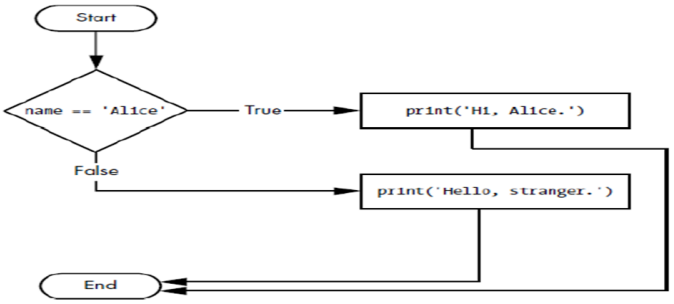
if else statement

•An if clause can optionally be followed by an else statement. The else clause is executed only when the if statement's condition is False.

➢ Example:

```
if name == 'Alice':
    print('Hi, Alice.')
else:
    print('Hello, stranger.')
```

➢ Flowchart:



The elif statement

•While only one of the if or else clauses will execute, we may have a case where we want one of many possible clauses to execute.

Ø The elif statement is an "else if" statement that always follows an if or another elif statement.

Ø It provides another condition that is checked only if all of the previous conditions were False.

•

➢ Example:

```
if name == 'Alice':
    print('Hi, Alice.')
elif age < 12:
    print('You are not Alice, kiddo.')
```

2.a	Develop a python program to check whether a number is Armstrong number or not.	[5]	CO1	L3

	# Python program to check if the number is an Armstrong number or not
	
	# take input from the user
	num = int(input("Enter a number: "))
	
	# initialize sum
	sum = 0
	
	# find the sum of the cube of each digit
	temp = num
	while temp > 0:
	  digit = temp % 10
	  sum += digit ** 3
	  temp //= 10
	
	# display the result
	if num == sum:
	  print(num,"is an Armstrong number")
	else:
	  print(num,"is not an Armstrong number")			

2.b	Explain the Scope of the variable. Differentiate local scope with global scope with example code snippets. ● Parameters and variables that are assigned in a called function are said to exist in that function's local scope. ● Variables that are assigned outside all functions are said to exist in the global scope. ● A variable that exists in a local scope is called a local variable, while a variable that exists in the global scope is called a global variable. ● A variable must be one or the other; it cannot be both local and global. ● When a scope is destroyed, all the values stored in the scope's variables are forgotten. ● There is only one global scope, and it is created when your program begins. When your program terminates, the global scope is destroyed, and all its variables are forgotten. ● A localscope is created whenever a function is called. Any variables assigned in this function exist within the local scope. ● When the function returns, the local scope is destroyed, and these variables are forgotten. ● Scopes matter for several reasons: 1. Code in the global scope cannot use any local variables. 2. However, a local scope can access global variables. 3. Code in a function's local scope cannot use variables in any other local scope. 4. We can use the same name for different variables if they are in different scopes. That is, there can be a local variable named spam and a global variable also named spam.	[5]	CO1	L2
3.a	Develop a python program to print following pattern: * * * * * * * * * * **Program:** n = int(input("enter number of lines::"))	[5]	CO1	L3

```
for i in range(1, n+1):
    # internal loop run for i times
    for k in range(1, i+1):
        print("*", end="")
    print()
```

3.b Compare while loop and for loop with example code snippets [5] CO1 L2

Feature	`for` Loop	`while` Loop
Initialization	Declared within the loop structure and executed once at the beginning.	Declared outside the loop; should be done explicitly before the loop.
Condition	Checked before each iteration.	Checked before each iteration.
Update	Executed after each iteration.	Executed inside the loop; needs to be handled explicitly.
Use Cases	Suitable for a known number of iterations or when looping over ranges.	Useful when the number of iterations is not known in advance or based on a condition.
Initialization and Update Scope	Limited to the loop body.	Scope extends beyond the loop; needs to be handled explicitly.

4.a Develop a program to compute Mean, Variance and Standard Deviation with [5] CO2 L3

	suitable messages. (Read N numbers from the console and create a list.)
	
	Program:
	
	import math as m
	n=int(input('Enter the number:'))
	lst=[]
	sum=0
	variance=0
	
	lst=(input('Enter the numbers with space:').split())
	
	for i in range(n):
	  sum+=float(lst[i])
	
	mean=sum/n
	
	for i in range(n):
	  variance+=(float(lst[i])-mean)**2
	
	variance=variance/n
	
	deviation=m.sqrt(variance)
	
	print('Mean value is:', mean)
	print("Variance is:", variance)
	print('Standard deviation is:', deviation)			

4.b	Illustrate how is tuple different from list and which function is used to convert list to tuple? Explain in detail.	[5]	CO2	L2

5.a	Explain Exception Handling in python with an example.	[6]	CO1	L2

	➢ If we don't want to crash the program due to errors instead we want the program to detect errors, handle them, and then continue to run.
	➢ For example,
	Program
	`def spam(divideBy):`
	`  return 42 / divideBy`
	`print(spam(2))`
	`print(spam(12))`
	`print(spam(0))`
	`print(spam(1))`
	Output
	`21.0`
	`3.5`
	`Traceback (most recent call last):`
	`  File "C:/zeroDivide.py", line 6, in <module>`
	`  print(spam(0))`
	`  File "C:/zeroDivide.py", line 2, in spam`
	`  return 42 / divideBy`
	`ZeroDivisionError: division by zero`
	➢ A ZeroDivisionError happens whenever we try to divide a number by zero. From the line number given in the error message, we know that the return statement in spam() is causing an error.
	➢ Errors can be handled with try and except statements.
	➢ The code that could potentially have an error is put in a try clause. The program execution moves to the start of a following except clause if an error happens.
	➢ We can put the previous divide-by-zero code in a try clause and have an except clause contain code to handle what happens when this error occurs.
	Program
	`def spam(divideBy):`
	`  try:`
	`  return 42 / divideBy`
	`  except ZeroDivisionError:`
	`  print('Error: Invalid argument.')`
	`print(spam(2))`
	`print(spam(12))`
	`print(spam(0))`
	`print(spam(1))`
	Output
	`21.0`
	`3.5`
	`Error: Invalid argument.`
	`None`
	`42.0`			

5.b	Develop a Python Program to display Fibonacci series of length
	
	Program:
	`n=int(input("enter the number"))`
	`a=0`
	`b=1`
	`sum=0`
	`i=1`
	`print("fibonacci series",end=' ')`
	`while(i<=n):`
	`  sum=a+b`
	`  print(a)`
	`  a=b`
	`  b=sum`
	`  i=i+1`
		[4]	CO1	L3

6.a	Explain Negative Indexing and slicing in List with suitable messages.	[6]	CO2	L2

Getting Sublists with Slices

- An index will get a single value from a list, a slice can get several values from a list, in the form of a new list.
- A slice is typed between square brackets, like an index, but it has two integers separated by a colon.
- **Difference between indexes and slices.**
  - spam[2] is a list with an index (one integer).
  - spam[1:4] is a list with a slice (two integers).
- In a slice, the first integer is the index where the slice starts. The second integer is the index where the slice ends (but will not include the value at the second index).

```
>>> spam = ['cat', 'bat', 'rat', 'elephant']
>>> spam[0:4]
['cat', 'bat', 'rat', 'elephant']
>>> spam[1:3]
['bat', 'rat']
>>> spam[0:-1]
['cat', 'bat', 'rat']
```

Negative Indexes

- We can also use negative integers for the index. The integer value -1 refers to the last index in a list, the value -2 refers to the second-to-last index in a list, and so on.

```
>>> spam = ['cat', 'bat', 'rat', 'elephant']
>>> spam[-1]
'elephant'
>>> spam[-3]
'bat'
>>> 'The ' + spam[-1] + ' is afraid of the ' + spam[-3] + '.'
'The elephant is afraid of the bat.'
```

6.b	Explain String Concatenation & String Replication with the help of example **String Concatenation:** + is used on two string values, it joins the strings as the string concatenation operator. If we try to use the + operator on a string and an integer value, Python will not know how to handle this, and it will display an error message. **eg: 'Alice' + 'Bob'** **'AliceBob'** **String Replication:** The * operator is used for multiplication when it operates on two integer or	[4]	CO2	L2

	floating-point values. But, when the * operator is used on one string value and one integer value, it becomes the string replication operator. The * operator can be used with only two numeric values (for multiplication) **eg: 'Alice' + 5** **'AliceAliceAliceAliceAlice'**			

7.a	Explain the following methods with suitable example:
	i) upper( ) ii) lower( ) iii) isupper( ) iv) islower( )
	
	i) upper() : converts string into upper case
	ii) lower() : converts string into lower case
	Eg:
	print('How are you?')
	feeling = input()
	if feeling.lower() == 'great':
	print('I feel great too.')
	else:
	print('I hope the rest of your day is good.')
	
	iii) isupper(): returns true if string passed inside function is in upper case
	iv) islower() : returns true if string passed inside function is in lower case
	Eg:
	>>> **spam = 'hello world!'**
	>>> **spam.islower()**
	True
	>>> **spam.isupper()**
	False
		[6]	CO3	L2

7.b	Explain the isX String methods with Example code snippet.
	•isalpha() returns True if the string consists only of letters and is not blank.
	>>>'hello'.isalpha()
	True
	>>> 'hello123'.isalpha()
	False
	
	•isalnum() returns True if the string consists only of letters and numbers and is not blank.
	>>> 'hello123'.isalnum()
	True
	>>> 'hello'.isalnum()
	True
		[4]	CO3	L2

	•isdecimal() returns True if the string consists only of numeric characters and is not blank.
	>>> '123'.isdecimal()
	True
	
	•isspace() returns True if the string consists only of spaces, tabs, and newlines and is not blank.
	>>> ‘ ’.isspace()
	True
	
	•istitle() returns True if the string consists only of words that begin with an uppercase letter followed by only lowercase letters.
	>>> 'This Is Title Case 123'.istitle()
	True
	>>> 'This Is not Title Case'.istitle()
	False			