**Practical No 2**

**PART A**

**A.1 AIM: - To implement control structure statements in Python.**

**A.2 Prerequisite**

Programming for problem solving and Object Oriented Programming

**A.3 Outcome**

After successful completion of this experiment, students will be able to understand and implement

1. Conditional block statements
2. Looping statements
3. Pass, Continue and Break statements

**A.4 Theory**

**Python - Decision Control: if, else, elif**

By default, statements in the script are executed sequentially from the first to the last. If the processing logic requires so, the sequential flow can be altered in two ways:

Conditional execution: a block of one or more statements will be executed if a certain expression is true.

Repetitive execution: a block of one or more statements will be repetitively executed as long as a certain expression is true.

**if Condition**

Python uses the if keyword to implement decision control. Python's syntax for executing a block conditionally is as below:

**Syntax**

**if [boolean expression]:**

statement1

statement2

...

Statement n

Any Boolean expression evaluating to True or False appears after the if keyword. Use the **:** symbol and press Enter after the expression to start a block with increased indent. One or more statements written with the same level of indent will be executed if the Boolean expression evaluates to True.

To end the block, decrease the indentation. Subsequent statements after the block will be executed out of the if condition. The following example demonstrates the if condition.

**Example**

>>>> if 10<100:

...   print("10 is less than 100")

**Output: 10 is less than 100**

**else Condition**

Along with the if statement, the else condition can be optionally used to define an alternate block of statements to be executed if the boolean expression in the if condition is not true.

**Syntax**

**if [boolean expression]:**

statement1

statement2

...

statementN

**else:**

statement1

statement2

...

statement

>>>> if 10>100:

...   print("10 is greater than 100")

... else:

...   print("10 is less than 100")

**Output: 10 is less than 100**

**elif Condition**

Use the elif condition is used to include multiple conditional expressions between if and else.

**Syntax**

if [boolean expression]:

[statements]

elif [boolean expresion]:

[statements]

elif [boolean expresion]:

[statements]

elif [boolean expresion]:

[statements]

else:

[statements]

>>>> x=10

>>>> if x==1:

...  print('X is 1')

... elif x==5:

...  print('X is 5')

... elif x==10:

...  print('X is 10')

... else:

...  print('X is something else')

**Output: X is 10**

**While Loop**

Loop is a very popular phrase in programming jargon. A program, by default, follows a sequential execution of statements. If the program flow is directed towards any of the earlier statements in the program, it constitutes a loop. However, sending it unconditionally causes an infinite loop, which is not desired.

Python uses the while and for keywords to constitute a conditional loop, by which repeated execution of a block of statements is done until a Boolean expression is true.

**Syntax**

**while [boolean expression]:**

statement1

statement2

...

statementN

**Example**

num =0

while num< 5:

num = num + 1

print("num =", num)

**For Loop**

Python's for keyword provides a more comprehensive mechanism to constitute a loop. The for loop is used with sequence types such as list, tuple and set. The body of the for loop is executed for each member element in the sequence. Hence, it doesn't require explicit verification of Boolean expression controlling the loop (as in the while loop).

**Syntax**

for x in sequence:

statement1

statement2

...

statementN

To start with, variable x in the for statement refers to the item at the 0 index in the sequence. The block of statements with increased uniform indent after the **:** symbol will be executed. Variable x now refers to the next item and repeats the body of the loop till the sequence is exhausted.

**range Function**

In Python, the range() method returns an immutable sequence of numbers. It can be used to control the repetition of a block in the for loop.

Syntax: range(**start index** , **stop index** , **step count** ).

for char in "Hello":

print (char)

>>> for x in range(0,10):

print(x)

>>> for x in range(10):

print(x)

>>> for x in range(1,10,3):

print(x)

The above example shows different variations in which for loop can be used in python. First example scans the entire string hello and prints it.

**Nested for Loop**

If a loop (for loop or while loop) contains another loop in its body block, we say that the two loops are nested. If the outer loop is designed to perform m iterations and the inner loop is designed to perform n repetitions, the body block of the inner loop will get executed m X n times.

for x in range(1,4):

for y in range(1,3):

print ("Hello World")

**else in Loop**

As you have learned before, the else clause is used along with the if statement. Python allows the else keyword to be used with the for and while loops too. The else block appears after the body of the loop. The statements in the else block will be executed after all iterations are completed. The program exits the loop only after the else block is executed.

Example:

for x in range(5):

print(x)

else:

print ("else block in loop")

print ("Out of loop")

**break Keyword**

The break keyword causes the abandonment of pending iterations of the current loop. The execution of the program jumps to the statement immediately after the body of the loop.

Example:

num=0

while num<5:

num=num+1

print (num)

if num==3:

break

print ("Out of loop")

**continue Keyword**

The effect of a continue statement is somewhat opposite to the break keyword. Instead of abandoning the pending iterations in the loop, the continue statement skips the remaining statements in the current loop and starts the next iteration.

Example

for num in range(1,6):

if num==3:

continue

print (num)

print ("Out of loop")

**pass Keyword**

The pass keyword as name suggests, does nothing. It is used as a dummy place holder whenever a syntactical requirement of a certain programming element is to be fulfilled without assigning any operation. In other words, the pass statement is simply ignored by the Python interpreter and can be seen as a null statement. It is generally used as a dummy statement in a code block, for example in the if or else block.

Example:

for num in range(1,6):

if num==3:

pass

else:

print (num)

**Task 1: Write a Python program to check whether a person is eligible for voting or not (accept age from user)**

**Task 2: Write a program to display “Hello” if a number entered by the user is a multiple of five, otherwise print “Bye”.**

**Task 3: Write a Python program to accept percentage from the user and display the grade according to following criteria:**

|  |  |
| --- | --- |
| **Marks** | **Grade** |
| **>90** | **A** |
| **>80 and <=90** | **B** |
| **>=60 and <=80** | **C** |
| **Below 60** | **D** |

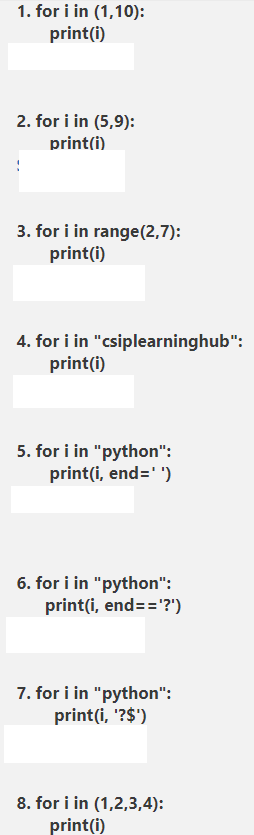
**Task 4: Write a program to accept the cost price of a bike and display the road tax to be paid according to following criteria.**

|  |  |
| --- | --- |
| **Cost Price (in Rs)** | **Tax** |
| **>100000** | **15%** |
| **>50000 and <=100000** | **10%** |
| **<=50000** | **5%** |

**Task 5: Write a Python Program to print first 10 natural numbers.**

**Task 6: Write a Python program to Read a Number n and Print the Sum of odd Natural Numbers between the range of 1 to n both inclusive**

**Task 7: Write the output of the following**

****

**Program 8: Write a Python program to find factorial of a number**

**Program 9: Write a python program to reverse a given number**

**Program 10: Write a python program to generate Fibonacci number series.**

**Program 11: Write a program to print all the strong numbers between 1 to 100**

**Note:** Strong number is a special number whose sum of the factorial of digits is equal to the original number. For Example: 145 is a strong number. Since, 1! + 4! + 5!

**Program 12: Write a Python program to print the following pattern:**

**5 4 3 2 1**

4 3 2 1

3 2 1

2 1

1

**Program 13: Accept the marks for the number of subjects studying in this semester. While accepting marks check the constraints that entered marks should not be negative as well as should not be more than 100 (if entered terminate the code). If the constraint is satisfied, calculate the total & percentage. If % is greater than equal to 92 display “Merit” if % is between 75 and 91 display “Distinction” if % is between 60 and 74 “First class” if % is between 45 to 59 display “Second class” else display “Fail”**

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

**(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Teams or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)**

|  |  |
| --- | --- |
| Roll No. C137 | Name: Naisha Shetty |
| Program: BTI | Division: D |
| Semester: 4 | Batch : D1 |
| Date of Experiment: 20/12/2023 | Date of Submission: 22/12/2023 |
| Grade : |  |

B.1 Software Code written by student:

**Inputs:**

**Program 1:**

age = int(input("Enter your age: "))

if age >= 18 :

print("You are eligible to vote !!")

else :

print("You are not eligible to vote.")

**Program 2:**

num = int(input("Enter a number: "))

if num % 5 == 0 :

print("Hello")

else :

print("Bye")

**Program 3:**

marks = int(input("Enter your marks: "))

if marks > 90 :

print("Grade A")

elif marks > 80 and marks <= 90 :

print("Grade B")

elif marks >= 60 and marks <= 80 :

print("Grade C")

else :

print("Grade D")

**Program 4:**

cost = int(input("Enter the cost price of the bike (in Rs): "))

if cost > 100000 :

print("15% Road Tax")

elif cost > 50000 and cost <= 100000 :

print("10% Road Tax")

else :

print("5% Road Tax")

**Program 5:**

print("The first 10 Natural Numbers are: ")

i = 1

while i <= 10 :

print(i)

i = i + 1

**Program 6:**

n = int(input("Enter a number : "))

oddsum = 0

while n >= 1 :

if n % 2 != 0 :

oddsum = oddsum + n

n = n - 1

else :

n = n - 1

print("The sum off all the odd numbers is ", oddsum)

**Program 7:**

7.1: 1

10

7.2: 5

9

7.3: 2

3

4

5

6

7.4: c

s

i

p

l

e

a

r

n

i

n

g

h

u

b

7.5: p y t h o n

7.6: ERROR

7.7: p ?$

y ?$

t ?$

h ?$

o ?$

n ?$

7.8: 1

2

3

4

**Program 8:**

num = int(input("Enter the number whose factorial you want to find : "))

factorial = 1

while num >= 1:

factorial = factorial \* num

num = num - 1

print("The factorial is ",factorial)

**Program 9:**

num = int(input("Enter a number: "))

reversed\_num = 0

while num != 0:

digit = num % 10

reversed\_num = reversed\_num \* 10 + digit

num //= 10

print("Reversed Number: " ,reversed\_num)

**Program 10:**

num = int(input("Enter the Fibonacci Number Range: "))

n1 = 0

n2 = 1

i = 0

while(i < num):

print(n1)

Next = n1 + n2

n1 = n2

n2 = Next

i = i + 1

**Program 11:**

import math

minimum = int(input("Enter the Minimum Value: "))

maximum = int(input("Enter the Maximum Value: "))

for number in range(minimum, maximum) :

temp = number

sum = 0

while(temp > 0) :

reminder = temp % 10

factorial = math.factorial(reminder)

sum = sum + factorial

temp = temp // 10

if (sum == number) :

print(number, "is a Strong Number.")

**Program 12:**

num = int(input("Enter the number of rows: "))

for row in range(num, 0, -1):

for column in range(num-row):

print(" ", end="")

for column in range(row, 0, -1):

print(column, end=" ")

print()

**Program 13:**

marks = []

subjects = int(input("Enter the number of subjects studying in this semester: "))

for i in range(subjects):

mark = int(input(f"Enter the mark for subject {i+1}: "))

if mark < 0 or mark > 100:

print("Invalid mark. Mark should not be negative and should not be more than 100.")

break

marks.append(mark)

total = sum(marks)

percentage = (total / (subjects \* 100)) \* 100

if percentage >= 92:

grade = "Merit"

elif percentage >= 75:

grade = "Distinction"

elif percentage >= 60:

grade = "First class"

elif percentage >= 45:

grade = "Second class"

else:

grade = "Fail"

print(f"Total Marks: {total}")

print(f"Percentage: {percentage}%")

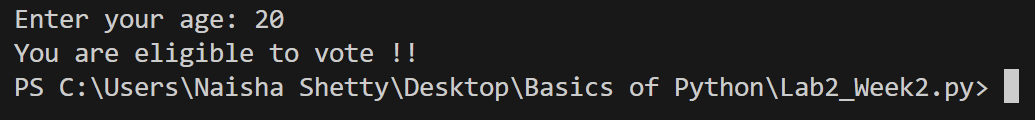
print(f"Grade: {grade}")

***(Paste your Python code completed during the 2 hours of practical in the lab here)***

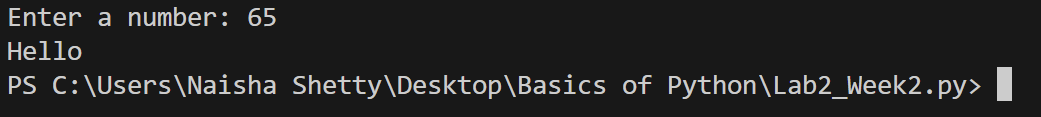
B.2 Input and Output:

**Outputs:**

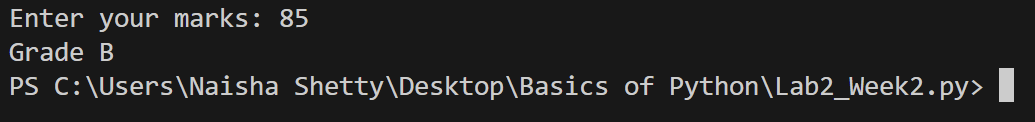
Program 1:



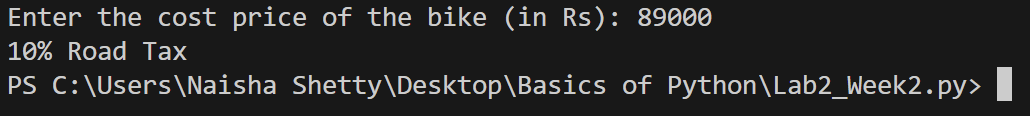
Program 2:



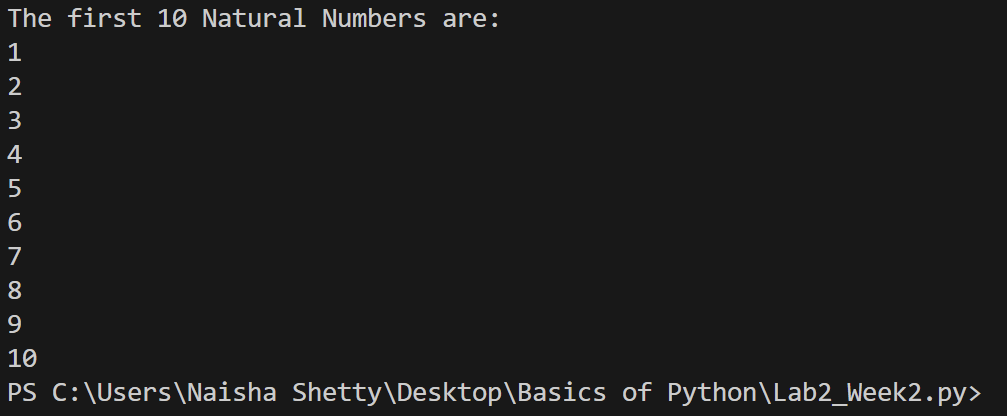
Program 3:



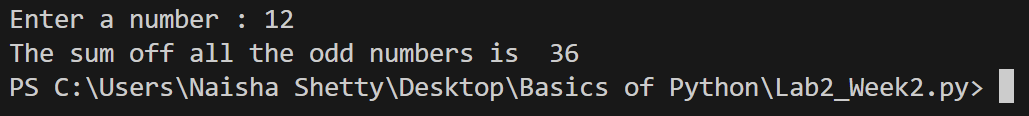
Program 4:



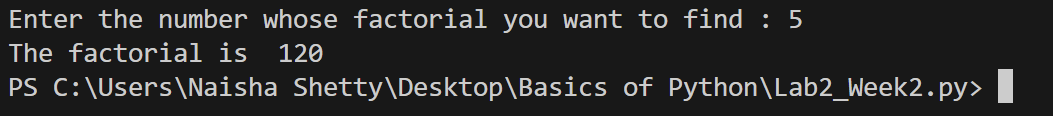
Program 5:



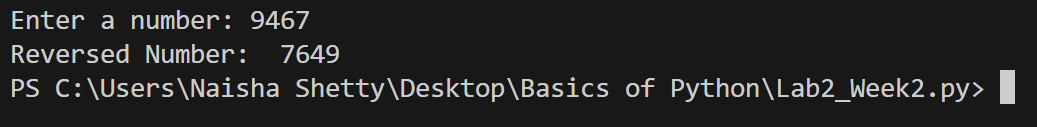
Program 6:



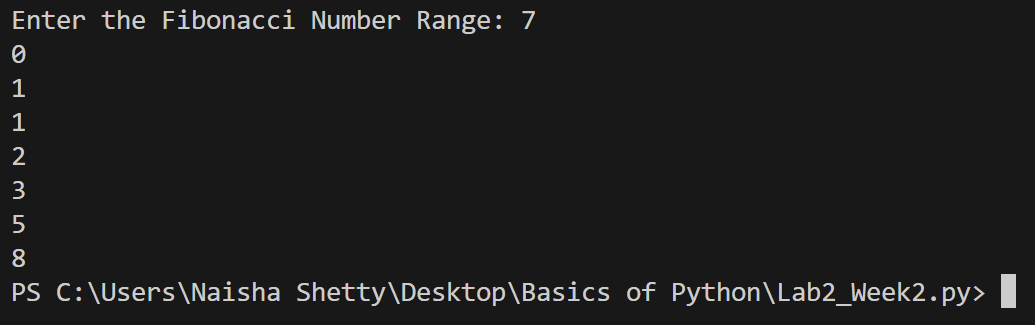
Program 8:



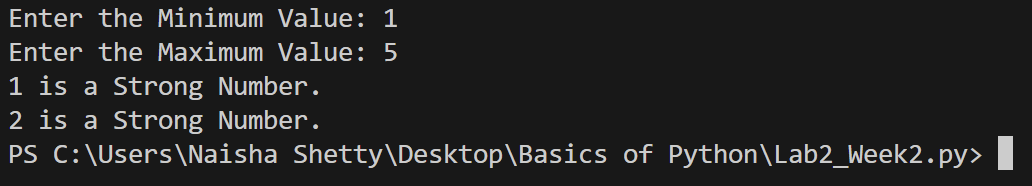
Program 9:



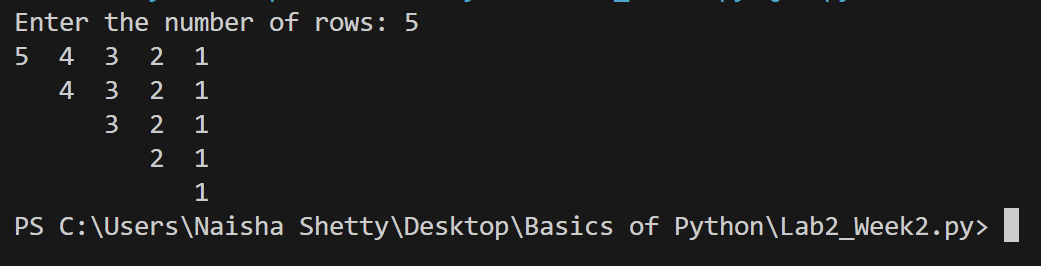
Program 10:



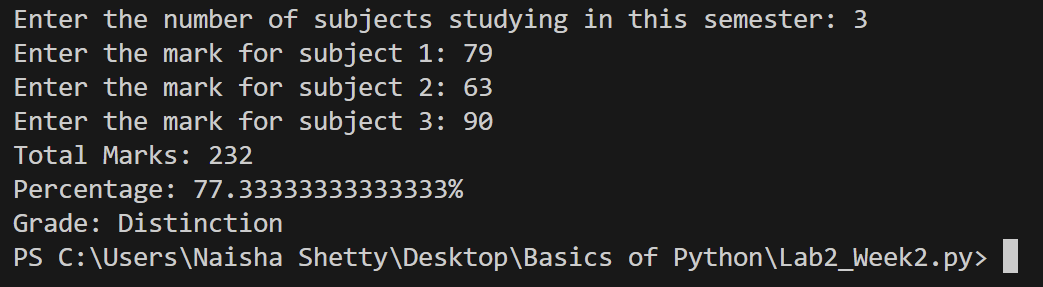
Program 11:



Program 12:



Program 13:



***(Paste your program input and output in following format. If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)***

B.3 Conclusion:

We learnt how to use the different types of loops in python.

*(****Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.1)***