Practical 1

# Objective: To be able to take input and save it in variable and be able to print out the variable’s value.

Syntax

variable = input()

print(variable)

## Concepts Used:

* input()
* print()

Practical 2

# Objective: To be able to use conditional statement.

Syntax

|  |  |  |  |
| --- | --- | --- | --- |
| variable = 12  if(variable > 18):  print(“Above 18”)  else:  print(“Below 18”) | variable = 12  if(variable < 18):  print(“Below 18”)  else:  print(“Above 18”) | variable = 12  if(variable == 18):  print(“Is 18”)  else:  print(“Is not 18”) | variable = 12  if(variable != 18):  print(“Is not 18”)  else:  print(“Is 18”) |

## Concepts Used:

* if()
* elif()
* else

Practical 3

# Objective: To be able to use loop.

Syntax: for loop in range

|  |  |  |
| --- | --- | --- |
| for i in range(11):  print(i) | for i in range(0, 11):  print(i) | for i in range(0,11,1):  print(i) |

Syntax: for loop in

arr = [11, 12, 3, 5]

for i in arr:

print(i)

Syntax: while loop

|  |  |  |  |
| --- | --- | --- | --- |
| i = 0  while( i < 11):  print(i)  i+=1 | i = 11  while( i > 0):  print(i)  i-=1 | i = 11  while( i == 11):  print(i) | i = 0  while( i !=11):  print(i)  i+=1 |

## Concepts Used:

* for
* in
* range()
* while()

Practical 4, 10

# Objective: To be able to create function.

Syntax

|  |  |
| --- | --- |
| def *is\_num1\_big* (num1, num2):  return num1>num2 | def *is\_num2\_big* (num1, num2):  return num1<num2 |

|  |  |
| --- | --- |
| def *is\_num1\_equals\_to\_num2* (num1, num2):  return num1==num2 | def *num1\_not\_equals\_to\_num2* (num1, num2):  return num1!=num2 |

|  |  |
| --- | --- |
| def *print\_sum*(num1, num2):  print(num1 + num2) | def *print\_sum*(num1, num2):  print(num1 + num2)  return |

## Concepts Used:

* def
* return

Practical 5, 6

# Objective: To be able to do recursion

Syntax

|  |  |
| --- | --- |
| def *print\_till\_zero*(num):  if(num < 0):  return  print(num)  return *print\_till\_zero*(num-1) | def *print\_till\_num*(num, increment = 0):  if(num < increment):  return  print(increment)  return *print\_till\_num*(num, increment+1) |

## Concepts Used:

* def
* return

Practical 7, 8, 9

# Objective: To be able to use array (list).

Syntax

|  |  |  |
| --- | --- | --- |
| arr = []  arr.append(1) | var = “abc”  arr = var.split() | arr = [1,3,4,6,2]  arr = arr.sort() |

## Concepts Used:

* list()
* split()
* sort()

Practical 11, 12

# Objective: To be able to implement object oriented programming.

Syntax

|  |  |
| --- | --- |
| class A:  def \_\_init\_\_(self, name):  self.name = name  def \_\_str\_\_(self):  return f'Your Name is {self.name}'  a = A(“abc”)  print(a) | class A:  def set\_name(self, name):  self.name = name  def get\_name(self):  return f'Your Name is {self.name}'  a = A()  a.set\_name("abc")  print(a.get\_name()) |

class A:

def set\_name(self, name):

self.name = name

def get\_name(self):

return f'Your Name is {self.name}'

class B(A):

def set\_address(self, address):

self.address = address

def get\_data(self):

return f'Your Name is {self.name} and your Address is {self.address}'

b = B()

b.set\_name("abc")

b.set\_address("A B C")

print(b.get\_data())

## Concept Used:

* class
* dunder functions
* inheritance

Practical 13

# Objective: To read and write text file using python.

Syntax

|  |  |
| --- | --- |
| text\_file = open("readme.txt","w")  text\_file.write("Hello\n")  text\_file.close() | text\_file = open("readme.txt","r")  print(text\_file.readline())  text\_file.close() |

## Concepts Used:

* open()
* close()
* write()
* readline()