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()