AEC-Python Programming

Assignment-1

1. Develop a python program to perform the bitwise operations on 60 and 13.

Program:

```
a=60  # 60 = 0011 1100
b=13  # 13 = 0000 1101
print("Bitwise Operations on 60 and 13:")
c=a&b
print("AND operation=",c)  # 12 = 0000 1100
d=a|b
print("OR operation=",d)   # 61 = 0011 1101
e=a^b
print("EXOR operation=",e) # 49 = 0011 0001
f=~a
g=~b
print("NOT operation on 60=",f)  # -61 = 1100 0011
print("NOT operation on 13=",g)  # -14 = 1111 0010
```

Output:

```
Bitwise Operations on 60 and 13:
AND operation= 12
OR operation= 61
```

EXOR operation= 49

NOT operation on 60= -61

NOT operation on 13= -14

2. Perform the following operations on the given list

A = ['abcd', 456 , 2.23, 'john', 70.2]

B = [123, 'john']

- Print list A and B

- Print the first element of List A

- Print the elements starting from 1-4 of list A

- Print the list B for two times

- Join the list A and B

- Replace all the elements of list B

- Delete the 'second' element from list A

- Insert the new element in the list B

Program:

```
print("*Peforming List Operations*")
A=['abcd',456,2.23,'john',70.2]
B=[123,'john']
print("A =",A)
print("B =",B)
print("First element of A =",A[0])
print("Elements of A from 1 to 4 =",A[1:])
print("Print B twice =",B*2)
print("Concatenate lists A and B =",A+B)
print("Replace B[0] in B =")
```

```
B[0]=789
print(B[0])
print("Replace B[1] in B =")
B[1]='Apple'
print(B[1])
print("Deleting second element of B =")
B.remove(B[1])
print("After deletion,B =",B)
print("Inserting an element in B =")
B.insert(0,888)
print("After insertion,B =",B)
```

Output:

```
*Peforming List Operations*
A = ['abcd', 456, 2.23, 'john', 70.2]
B = [123, 'john']
First element of A = abcd
Elements of A from 1 to 4 = [456, 2.23, 'john', 70.2]
Print B twice = [123, 'john', 123, 'john']
Concatenate lists A and B = ['abcd', 456, 2.23, 'john', 70.2, 123, 'john']
Replace B[0] in B =
789
```

Replace B[1] in B =

Apple

Deleting second element of B =

After deletion,B = [789]

Inserting an element in B =

After insertion,B = [888, 789]

3. Develop a python program that prints the reference of the variables a=10 and b=10 using the identity operator.

Program:

```
a=10

b=10

print("Reference comparison using Identity Operators:")

print(a is b) #TRUE

print(a is not b) #FALSE
```

Output:

```
Reference comparison using Identity Operators:

True

False
```

---

4. Develop a python program on the nested list.

Program:

```
A=[[1,2,3],[4,5,6],[7,8,9]]

print("Nested List Operations")

print("The List A =",A)

print("First Element of A =",A[0])

print("Second Element of A =",A[1])

print("Third Element of A =",A[2])
```

Output:

```
Nested List Operations

The List A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

First Element of A = [1, 2, 3]

Second Element of A = [4, 5, 6]

Third Element of A = [7, 8, 9]

-------------------------------------------------------------------------------------------------------------
-------------------------
```

5. Illustrate how list is different from the tuples.

Program:

```
print("Tuples cannot be manipulated during runtime(they are immutable),but Lists are mutable")

t=(1,2,3,4)

l=[11,22,33,44]

print("Tuple =",t)

print("List =",l)

print("*Operations on Tuple*")

c=t.count(4)

print("Count of the element '4' =",c)

t2=(t+(5,6,7))

print("Appending values to Tuple t=",t2)

print("Minimum value of Tuple t=",min(t))

print("Maximum value of Tuple t=",max(t))

print("Length of Tuple t=",len(t))
```

Output:

```
Tuples cannot be manipulated during runtime(they are immutable),but Lists are mutable

Tuple = (1, 2, 3, 4)

List = [11, 22, 33, 44]

*Operations on Tuple*
```

Count of the element '4' = 1

Appending values to Tuple t= (1, 2, 3, 4, 5, 6, 7)

Minimum value of Tuple t= 1

Maximum value of Tuple t= 4

Length of Tuple t= 4

6. Develop a python program to calculate the gross salary of the employee and provide the bonus of Rs. 5000 to all the employees having their basic salary greater than Rs. 20000

Program:

```
bs=int(input("Enter your salary: "))
da=(25/100)*bs
hra=(15/100)*bs
if(bs>=20000):
    gs=da+hra+bs+5000
    print("Congratulations! You have received a Bonus of Rs.5000!")
else:
    gs=da+hra+bs
    print("Sorry! You have not received any Bonus")
print("Gross Salary =",gs)
```

Output:

a) Enter your salary: 10000

Sorry! You have not received any Bonus

Gross Salary = 14000.0

b) Enter your salary: 25000

Congratulations! You have received a Bonus of Rs.5000!

Gross Salary = 40000

7. Develop a python program that apply the discount of 15% on purchase of items above Rs. 1000.

Program:

```
cost=int(input("Enter your Bill amount : "))
if(cost>=1000):
    discount=(15/100)*cost
    total=cost-discount
    print("Congratulations! You have received a Discount of 15%!")
else:
    total=cost
    print("Sorry! You have not received any Discount")
print("Total Bill amount =",total)
```

Output:

a) Enter your Bill amount : 1200

Congratulations! You have received a Discount of 15%!

Total Bill amount = 1020.0

b) Enter your Bill amount : 850

Sorry! You have not received any Discount

Total Bill amount = 850

8. Develop a python program to calculate the SGPA for the courses of IV Semester

Program:

```
print("*SGPA of 4th Semester Subjects*")
print("Enter the marks for each of the following subjects(out of 100):")
jp=int(input("JAVA Programming :"))
ada=int(input("Analysis and Design of Algorithms :"))
mc=int(input("MicroController and Embedded Systems :"))
dmgt=int(input("Discrete Mathematics and Graph Theory :"))
print("--------------------------------")
print("Enter the credits for each subject:")
a=int(input("JAVA Programming :"))
b=int(input("Analysis and Design of Algorithms :"))
c=int(input("MicroController and Embedded Systems :"))
d=int(input("Discrete Mathematics and Graph Theory :"))
creds=a+b+c+d
jp_w=a*jp
ada_x=b*ada
mc_y=c*mc
dmgt_z=d*dmgt
sgpa=(jp_w+ada_x+mc_y+dmgt_z)/creds
final=sgpa/10
```

```
print("------------------------------")
print("SGPA of the Student =",final)
```

Output:

```
*SGPA of 4th Semester Subjects*
Enter the marks for each of the following subjects(out of 100):
JAVA Programming :90
Analysis and Design of Algorithms :98
MicroController and Embedded Systems :89
Discrete Mathematics and Graph Theory :80
--------------------------------
Enter the credits for each subject:
JAVA Programming :3
Analysis and Design of Algorithms :3
MicroController and Embedded Systems :3
Discrete Mathematics and Graph Theory :3
--------------------------------
SGPA of the Student = 8.925
------------------------------------------------------------------------------------------------------------
------------------------
```

9. Develop a python program to find the largest among three numbers.

Program:

```
print("Enter Three numbers:")
a=int(input())
b=int(input())
c=int(input())
if(a>b) and (a>c):
   print(a," is the largest")
elif(b>a) and (b>c):
    print(b," is the largest")
else:
    print(c," is the largest")
```

Output:

```
Enter Three numbers:
20
30
10
30  is the largest
```

---

10. Write a python program to calculate the gross salary of 10 employees

Program:

```
print("Gross Salary for 10 Employees:-")
i=1.
for i in range(10):
    print("Enter your salary - Employee ",i+1)
    bs=int(input())
    da=(25/100)*bs
    hra=(15/100)*bs
    gs=da+hra+bs
    print("Gross Salary of Employee ",i+1)
    print(gs)
    print("------------------------")
print("End of Calculation")
```

Output:

```
Gross Salary for 10 Employees:-
Enter your salary - Employee  1
1000
Gross Salary of Employee  1
1400.0
-----------------------
```

Enter your salary - Employee 2

1200

Gross Salary of Employee 2

1680.0

Enter your salary - Employee 3

1400

Gross Salary of Employee 3

1960.0

Enter your salary - Employee 4

1600

Gross Salary of Employee 4

2240.0

Enter your salary - Employee 5

1800

Gross Salary of Employee 5

2520.0

Enter your salary - Employee 6

2000

Gross Salary of Employee 6

2800.0

Enter your salary - Employee 7

2200

Gross Salary of Employee 7

3080.0

Enter your salary - Employee 8

2400

Gross Salary of Employee 8

3360.0

Enter your salary - Employee 9

2600

Gross Salary of Employee 9

3640.0

Enter your salary - Employee 10

2800

Gross Salary of Employee 10

3920.0

End of Calculation

---

11. Write a python program that prints the sum of all even and odd numbers that are between two given numbers.

Program:

a)Displaying the list of Odd and Even Numbers:

```
print("Enter the First and Last number:")
a=int(input())
b=int(input())
c=a
d=b
total_even=0
total_odd=0
if a>b:
   print("Even Numbers:")
   while(a>b):
      if b%2==0:
         print(b)
         total_even=total_even+b
      b=b+1
else:
   print("Even Numbers:")
   while(a<b):
      if a%2==0:
```

```
            print(a)
            total_even=total_even+a
        a=a+1
if c>d:
    print("Odd Numbers:")
    while(c>d):
        if d%2!=0:
            print(d)
            total_odd=total_odd+c
        d=d+1
else:
    print("Odd Numbers:")
    while(c<d):
        if c%2!=0:
            print(c)
            total_odd=total_odd+d
        c=c+1
print("Sum of EVEN NUMBERS =",total_even)
print("Sum of ODD NUMBERS =",total_odd)
```

Output:

```
Enter the First and Last number:
```

20

10

Even Numbers:

10

12

14

16

18

Odd Numbers:

11

13

15

17

19

Sum of EVEN NUMBERS = 70

Sum of ODD NUMBERS = 100

b)Without Displaying the numbers

```
print("Enter the First and Last number:")
```

```
a=int(input())
b=int(input())
total_even=0
total_odd=0
if a>b:
   while(a>b):
      if b%2==0:
         total_even=total_even+b
         total_odd=total_odd+a
      b=b+1
else:
   while(a<b):
      if a%2==0:
         total_even=total_even+a
         total_odd=total_odd+b
      a=a+1

print("Sum of EVEN NUMBERS =",total_even)
print("Sum of ODD NUMBERS =",total_odd)
```

Output:

Enter the First and Last number:

10

20

Sum of EVEN NUMBERS = 70

Sum of ODD NUMBERS = 100

---

12. With example program demonstrate the working of Range function.

Program:

```
print("Program to demonstrate the use of 'range()' by printing numbers from 1 to 20:")

x=range(1,21)

for i in x:

    print(i)
```

Output:

```
Program to demonstrate the use of 'range()' by printing numbers from 1 to 20:
1
2
3
4
5
6
7
8
9
```

10

11

12

13

14

15

16

17

18

19

20

Assignment-2

1. **Develop a python program that demonstrate the working of a function with arguments and with return value**

   Program:

```
def add_numbers(x, y):
    result = x + y
    return result

result = add_numbers(5, 3)
print("Sum:", result)
```

   Output:

```
Sum:8
```

2. **With Python program illustrate the function calling another function**

Program:

```
def greet(name):
    return "Hello, " + name + "!"

def welcome(name):
    return "Welcome, " + name + "!"

message = greet("Alice") + " " + welcome("Bob")
print(message)
```

Output:

```
Hello, Alice! Welcome, Bob!
```

---

3. **Write a Python program that illustrate the working of function without argument and with return value**

Program:

```
def get_pi():
    return 3.14159

pi = get_pi()
print("The value of pi is:", pi)
```

Output:

```
The value of pi is:3.14159
```

---

4. **Write python program that demonstrate the function with positional arguments**

Program:

```
def calculate_power(base, exponent):
    result = base ** exponent
    return result

power = calculate_power(2, 3)
print("Result:", power)
```

Output:

Result: 8

---

5. **Write python program that demonstrate the function with default arguments**

Program:

```
def greet(name, message="Hello"):
    return message + ", " + name + "!"

greeting = greet("Alice")
print(greeting)
```

Output:

Hello,Alice!

---

6. **Illustrate the process of creating an object using python program**

Program:

```
class MyClass:
    def __init__(self, name):
        self.name = name

obj = MyClass("John")
print("Object's name:", obj.name)
```

Output:

Object's name: John

---

7. **Demonstrate the working of empty class**

Program:

```
class EmptyClass:
    pass
```

Output:

No specific output for an empty class definition

8**. Define a class student and with help of object define the function getdata() and displaydata()**

Program:

```
class Student:
    def getdata(self, name, roll):
        self.name = name
        self.roll = roll

    def displaydata(self):
        print("Name:", self.name)
        print("Roll:", self.roll)

student_obj = Student()
student_obj.getdata("Alice", 101)
student_obj.displaydata()
```

Output:

```
Name: Alice
Roll: 101
```

9.**Write a python program to find the area of rectangle**

Program:

```
def calculate_area(length, width):
    area = length * width
    return area
```

```
length = 5
width = 3
area = calculate_area(length, width)
print("Area of the rectangle:", area)
```

Output:

```
Area of the rectangle: 15
```

---

10. Analysze the importance of 'Self' as a default positional argument in functions

In Python, `self` is a convention used in class methods to refer to the instance itself, allowing methods to access and manipulate instance-specific data, ensuring encapsulation, inheritance, and polymorphism, making code readable and facilitating OOP principles.

Program:

```
class MyClass:
    def _init_(self, value):
        self.value = value

    def increment(self, amount):
        self.value += amount
```

```
obj1 = MyClass(5)
obj2 = MyClass(10)

obj1.increment(2)
print(obj1.value)
print(obj2.value)
```

Output:

Output: 7
Output: 10

---

11. Develop a python program that demonstrate the working of _init_() function

Program:

```
class MyClass:
    def __init__(self, value):
        self.value = value

obj = MyClass(42)
print("Value:", obj.value)
```

Output:

Value: 42

---

12. Develop a python program that assigns the values to the attributes of student class using _init_() function

Program:

```
class Student:
    def __init__(self, name, roll):
        self.name = name
        self.roll = roll

student_obj = Student("Bob", 201)
print("Name:", student_obj.name)
print("Roll:", student_obj.roll)
```

Output:

Name: Bob

Roll: 201
