Q1

as the
entity

- ○ An identity can be changed once created.
- ○ Two strings with same value point to different locations.
- ◉ id(object) is unique and Constant for an object during its lifetime.

Q2

nsole,

IdenopExample1.py

```
1  n1=int(input("Enter an integer: "))
2  n2=int(input("Enter the same integer: "))
3
4  print("x is y {0}".format(n1 is n2))
5  print("x is not y {0}".format(n1 is not n2))
6
7  n1=float(input("Enter a Float Number: "))
8  n2=float(input("Enter the same Number: "))
9
10 print("x is y {0}".format(n1 is n2))
11 print("x is not y {0}".format(n1 is not n2))
```

Q3

ıme

IdenopExample2.py

```
1  x=int(input("x: "))
2  y=int(input("y: "))
3  print("{0} is {1} {2}".format(x,y,x is y))
```

Q4

e same

IdenopExample3.py

```
1  x=int(input("x: "))
2  y=int(input("y: "))
3
4  print("{0} is not {1} {2}".format(x,y,x is not y))
```

Q1

iority.

' and

- [x] Operator precedence is performed based on priority of an operator.
- [x] Associativity is used when two operators of same precedence are in the same expressic
- [ ] Associativity is only from Left to Right .
- [ ] The operator precedence is only used on Arithmetic operators.

Q3

OppreceExample1.py

```
1  a=int(input("a: "))
2  b=int(input("b: "))
3
4  print("{0} + {1} * {2} = {3}".format(a,b,5,a+b*5))
5  print("{0} + {1} * {2} / 2 = {3}".format(a,b,6,a+b*6/2))
```

Q4

tion.

OppreceExample2.py

```
1  a=int(input("a: "))
2  b=int(input("b: "))
3
4  print("{0} + {1} * {2} = {3}".format(a,b,5,a+b*5))
5  print("{0} + {1} * {2} * 10 / 2 = {3}".format(a,b,5,a+b*5*10/2))
6
```

Q5

OppreceExample3.py

```
1  a=int(input("a: "))
2  b=int(input("b: "))
3  c=int(input("c: "))
4
5  print("a and b or c {0}".format(a and b or c))
6  print("a or b and c {0}".format(a or b and c))
```

Q6

OppreceExample4.py

```
a=int(input("a: "))
b=int(input("b: "))
c=int(input("c: "))

print("{0} and {1} and {2} or {3} is {4}".format(a,b,c,a,a and b and c or a))
print("{0} or {1} and {2} and {3} is {4}".format(a,b,c,a,a or b and c and a))
```