

Week3 Set7

TANISHA BISHT
RA1911003010259

Q1

Write a python program to count number of objects in a class

```
In [3]: class Person:
        count = 0
        def __init__(self, name, age):
            self.name = name
            self.age = age
            Person.count += 1
            print('The age of ' + name + ' is ' + str(age))

student1 = Person('Tanisha', 19)
student2 = Person('Sudhi', 22)
student3 = Person('Prak', 18)
student4 = Person('Eeshu', 15)
student5 = Person('Shiv', 21)

print('\nThe number of objects of the class person is: ' + str(Person.count))

The age of Tanisha is 19
The age of Sudhi is 22
The age of Prak is 18
The age of Eeshu is 15
The age of Shiv is 21

The number of objects of the class person is: 5
```

Q2

Create a Vehicle class with max_speed and mileage instance attributes.

```
In [7]: class Vehicle:
        def __init__(self, max_speed, mileage):
            self.max_speed = max_speed
            self.mileage = mileage
            print('The max speed is: ' + str(max_speed) + ' km/h')
            print('The mileage is: ' + str(mileage) + ' kmpl')

vehicle1 = Vehicle(338.8, 32.26)

The max speed is: 338.8 km/h
The mileage is: 32.26 kmpl
```

Q3

Create child class Bus that will inherit all of the variables and methods of the Vehicle class.

```
In [15]: class Vehicle:
        def __init__(self, max_speed, mileage):
            self.max_speed = max_speed
            self.mileage = mileage
            print('The max speed is: ' + str(max_speed) + ' km/h')
            print('The mileage is: ' + str(mileage) + ' kmpl')

        class Bus(Vehicle):
            pass

bus1 = Bus(338.8, 32.26)

The max speed is: 338.8 km/h
The mileage is: 32.26 kmpl
```

Q4

Create a Btech_SecondYear_CS class where the constructor function sorts the student name based on their percentages of attendance (using quick sort). After displaying the student list, the destructor function will be called for free the memory.

```
In [18]: def partition(arr, low, high):
        i = (low-1)
        pivot = arr[high]
        for j in range(low, high):
            if arr[j] <= pivot:
                i = i+1
                arr[i], arr[j] = arr[j], arr[i]
        arr[i+1], arr[high] = arr[high], arr[i+1]
        return i+1

def quickSort(arr, low, high):
    if len(arr) == 1:
        return arr
    if low < high:
        pi = partition(arr, low, high)
        quickSort(arr, low, pi-1)
        quickSort(arr, pi+1, high)

class Btech_SecondYear_CS:
    def __init__(self, studentList):
        dicLen = len(studentList)

        studentPercentage = list(studentList.values())
        quickSort(studentPercentage, 0, dicLen-1)

        key_list = list(studentList.keys())
        value_list = list(studentList.values())

        studentNames = []
        for percent in studentPercentage:
            pos = value_list.index(percent)
            studentNames.append(key_list[pos])
        self.names = studentNames

        # self.names = [name for name,per in sorted(studentList.items(), key=lambda x:x[1])]
        print('Names of students arranged in ascending order according to their percentage is -')
        print(self.names)

    def __del__(self):
        print('\nDestructor called, Btech_SecondYear_CS deleted.')

dic = {
    'Tanisha' : 98,
    'Prakk' : 100,
    'Sudhi' : 91,
    'Eeshu' : 60
}

CSStudents = Btech_SecondYear_CS(dic)
del CSStudents

Destructor called, Btech_SecondYear_CS deleted.
Names of students arranged in ascending order according to their percentage is -
['Eeshu', 'Sudhi', 'Tanisha', 'Prakk']

Destructor called, Btech_SecondYear_CS deleted.
```

Q5

Write a python program to implement queue structure using class

```
In [1]: class Queue:
        def __init__(self):
            self.items = []

        def is_empty(self):
            return self.items == []

        def enqueue(self, data):
            self.items.append(data)

        def dequeue(self):
            return self.items.pop(0)

q = Queue()
while True:
    print('1. enqueue <value>')
    print('2. dequeue')
    print('3. display')
    print('4. quit')
    num = int(input('Enter the number. What would you like to do? '))

    if num == 1:
        value = input('Enter the value you want to add in the queue')
        q.enqueue(value)
        continue

    elif num == 2:
        if q.is_empty():
            print('Queue is empty.')
        else:
            print('The value that is dequeued is: ' + str(q.dequeue()))
            continue

    elif num == 3:
        print('The queue is: ')
        print(q.items)

    else:
        break

1. enqueue <value>
2. dequeue
3. display
4. quit
Enter the number. What would you like to do? 1
Enter the value you want to add in the queue12
1. enqueue <value>
2. dequeue
3. display
4. quit
Enter the number. What would you like to do? 1
Enter the value you want to add in the queue10
1. enqueue <value>
2. dequeue
3. display
4. quit
Enter the number. What would you like to do? 3
The queue is:
['12', '10']
1. enqueue <value>
2. dequeue
3. display
4. quit
Enter the number. What would you like to do? 4
```