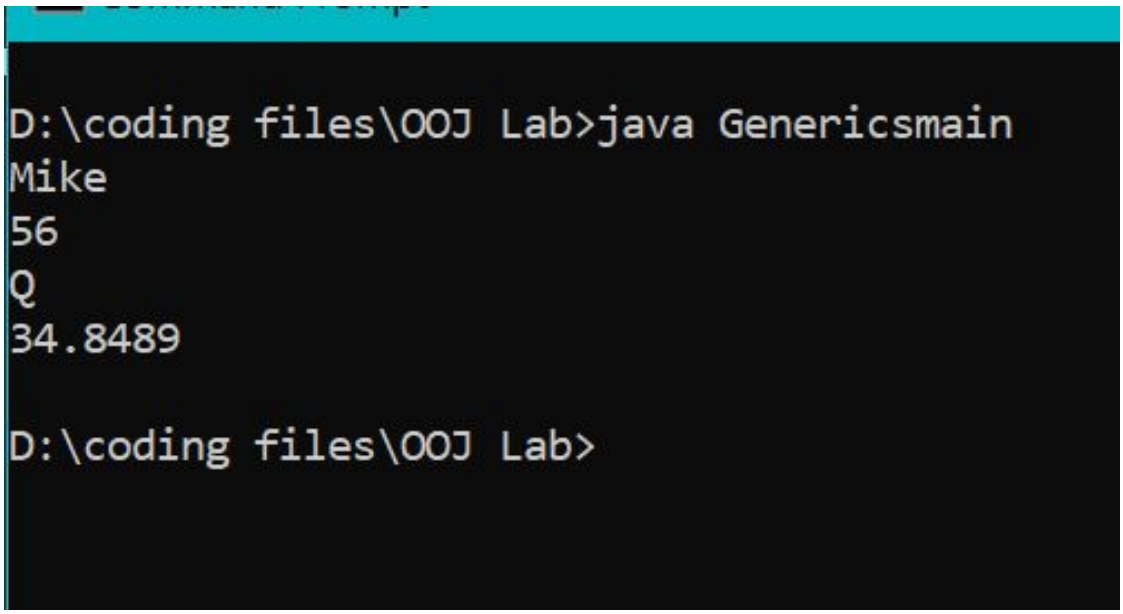
LAB - 7 & 8

/*Write a program to demonstrate generics with multiple object parameters.*/

```
class myGen<a,b>{
    a obj1;
    b obj2;

  myGen(a obj1, b obj2){
      this.obj1 = obj1;
      this.obj2 = obj2;
  }
  void Display(){
      System.out.println(obj1);
      System.out.println(obj2);
  }
}
class Genericsmain{
    public static void main(String args[]){
        myGen<String,Integer>myG1 = new myGen<String,Integer>("Mike",56);
        myGen<Character,Double>myG2 = new myGen<Character,Double>('Q',34.8489);
        myG1.Display();
        myG2.Display();
    }
}
```



**/*Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class
called Father and derived class called Son which extends the base class. In Father class, implement a constructor which takes the age and throws the exception Wrong Age( ) when the input
age<0. In Son class, implement a constructor that cases both father and son's age and throws an
exception if Son's age is >= father's age*/**

```
import java.util.Scanner;
class WrongAge extends Exception{
  public WrongAge(String s){
    super(s);
  }
}
class Father{
  int fatherAge;
  int sonAge;
  Father(int fAge,int sAge) throws WrongAge{
    if(fAge == sAge){
      throw new WrongAge("Father's age is equal to son's age");
    }
    else{
      this.fatherAge = fAge;
      this.sonAge = sAge;
    }
  }
}
class Son extends Father{
  Son(int fAge, int sAge) throws WrongAge{
    super(fAge,sAge);
    if(sAge>=fAge){
      throw new WrongAge("Son's age is qual to or greater than father's age");
    }
  }
  void Display(){
    System.out.println("Father's age:"+fatherAge);
    System.out.println("Son's age:"+sonAge);
  }
}
```

```
class Wrongage_main{
  public static void main(String[] args){
    int fAge,sAge;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter Father's Age: ");
    fAge = sc.nextInt();
    System.out.println("Enter Son's age: ");
    sAge = sc.nextInt();
    try{
      Son son = new Son(fAge,sAge);
      son.Display();
    }catch(WrongAge err){
      System.out.println("Exception "+err);
    }
  }
}
```

```
D:\coding files\OOJ Lab>javac wrongage_exception.java

D:\coding files\OOJ Lab>java Wrongage_main
Enter Father's Age:
24
Enter Son's age:
3
Father's age:24
Son's age:3

D:\coding files\OOJ Lab>java Wrongage_main
Enter Father's Age:
13
Enter Son's age:
45
Exception WrongAge: Son's age is qual to or greater than father's age

D:\coding files\OOJ Lab>
```