Date: 20/11/2024

~~8)~~ ~~Creating your own exception subclasses~~

~~Define a subclass of Exception the Exception~~

8) Write a program that demonstrates handling of exceptions in inheritance ~~tree~~ tree Create a base class called "Father" and derived class called as "Son" which extends the base class In Father's class implement a constructor which takes the age and throws the exception WrongAge() when the input age is less than zero In son's class implement a constructor that uses father and son's age and throws an exception if son's age greater than or equal to father's age

→
```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}
class SonAgeException extends Exception {
    public SonAgeException(String message) {
        super(message);
    }
}
class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
        if (age <= 0) {
            throw new WrongAgeException("Father's age cannot
                be negative or zero");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}
```

```
class Son extends Father {
    private int SonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeExcep.
                                        SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be
        greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}
public class fatherSon {
    public static void main(String args[]) {
        while (true) {
            Scanner sc = new Scanner(System.in);
            try {
                System.out.println("Enter Father's Age: ");
                int fatherAge = sc.nextInt();
                Father father = new Father(fatherAge);
                System.out.println("Enter's Son's Age: ");
                int sonAge = sc.nextInt();
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Succesfully");
            }
            catch(WrongAgeException e) {
                System.out.println(e.getMessage());
            }
            catch(SonAgeException e) {
                System.out.println(e.getMessage());
            }
            System.out.println("Would you like to re enter
                                    details(Y/N):");
            String input = sc.next();
```

```
        if (input.equalsIgnoreCase("n")) {
            break;
        }
    }
    System.out.println("Name : Akshay.S");
    System.out.println("USN: 1BM23CS022");
  }
}
```

Output:

```
Enter Father's Age: -1
Father's age cannot be negative or zero
Would you like to re-enter details (Y/N):
Y
Enter Father's Age: 30
Enter Son's Age: 35
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/N):
Y
Enter Father's Age: 30
Enter Son's Age: 5
Accepted Successfully
Would you like to re-enter details (Y/N):
N
Name: Akshay.S
USN: 1BM23CS022
```