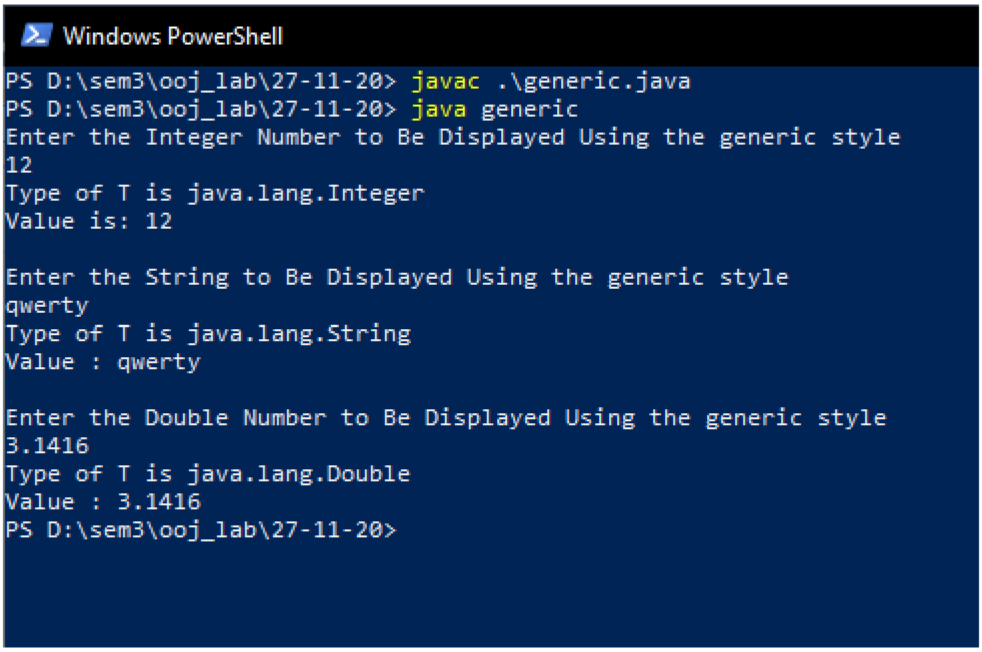
Program 7

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

```
import java.io.*;
import java.lang.*;
import java.util.*;

class gen<T>
{
      T ob;
      gen(T o)
      {
            ob=o;
      }
      T getob()
      {
            return ob;
      }
      void showtype()
      {
            System.out.println("Type of T is " +
ob.getClass().getName());
      }
}

class generic
{
      public static void main(String[] args)
      {
            String n;
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter the Integer Number to Be
Displayed Using the generic style");
            n=sc.next();
            gen<Integer> ob1=new gen<Integer>(Integer.parseInt(n));
            ob1.showtype();
            int val=ob1.getob();
            System.out.println("Value is: " + val);

            System.out.println();

            System.out.println("Enter the String to Be Displayed
Using the generic style");
            n=sc.next();
            gen<String> ob2=new gen<String>(n);
            ob2.showtype();
            String x=ob2.getob();
            System.out.println("Value : " + x);
```

```
                System.out.println();


                System.out.println("Enter the Double Number to Be
Displayed Using the generic style");
                n=sc.next();
                gen<Double> ob3=new gen<Double>(Double.parseDouble(n));
                ob3.showtype();
                double ans=ob3.getob();
                System.out.println("Value : " + ans);

        }
}
```

Output



```
PS D:\sem3\ooj_lab\27-11-20> javac .\generic.java
PS D:\sem3\ooj_lab\27-11-20> java generic
Enter the Integer Number to Be Displayed Using the generic style
12
Type of T is java.lang.Integer
Value is: 12

Enter the String to Be Displayed Using the generic style
qwerty
Type of T is java.lang.String
Value : qwerty

Enter the Double Number to Be Displayed Using the generic style
3.1416
Type of T is java.lang.Double
Value : 3.1416
PS D:\sem3\ooj_lab\27-11-20>
```

****

Program 8

2. 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 WrongAge( ) when the input age<0. In Son class, implement a constructor

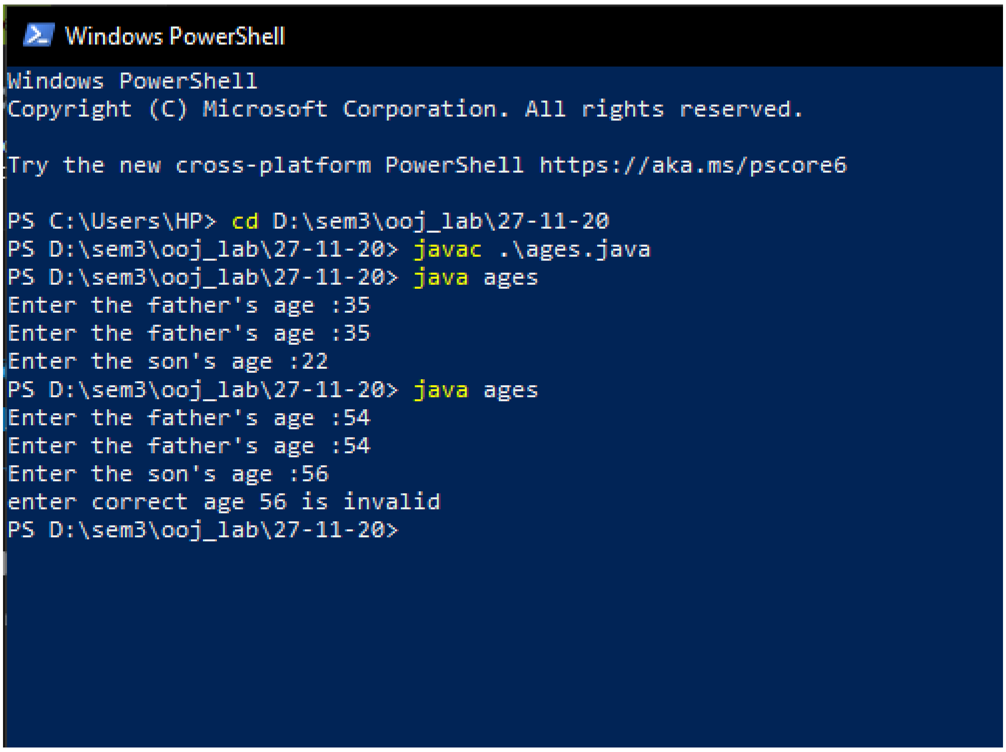
that cases both father and son’s age and throws an exception if son’s age is >=father’s

age.

```
import java.util.*;

class WrongAge extends Exception {

 int detail;

 WrongAge(int a) {

  detail = a;

 }

 public String toString() {

  return "enter correct age "+detail+" is invalid";

 }


}

class father{

 public int age;

 Scanner in =new Scanner(System.in);

 father() throws WrongAge{

  System.out.print("Enter the father's age :");

  age= in.nextInt();

  if(age<0)

    throw new WrongAge(age);

 }

}

class son extends father{

 Scanner in =new Scanner(System.in);

 int fage;

 son(father f) throws WrongAge{

  this.fage=f.age;

  System.out.print("Enter the son's age :");

  this.age= in.nextInt();
```

```
  if(this.age<0)
    throw new WrongAge(age);
  if(this.age>f.age)
    throw new WrongAge(age);
 }

}
class ages{
 public static void main(String[] args){
  try{
   father f= new father();
   son s= new son(f);
  }
  catch(Exception e){
    System.out.println(e);
 }
 }
}
```

Output:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\HP> cd D:\sem3\ooj_lab\27-11-20
PS D:\sem3\ooj_lab\27-11-20> javac .\ages.java
PS D:\sem3\ooj_lab\27-11-20> java ages
Enter the father's age :35
Enter the father's age :35
Enter the son's age :22
PS D:\sem3\ooj_lab\27-11-20> java ages
Enter the father's age :54
Enter the father's age :54
Enter the son's age :56
enter correct age 56 is invalid
PS D:\sem3\ooj_lab\27-11-20>
```

****