

OOJ LAB-WEEK 10-LAB PROG 6 AND 7

Program and Output

Mallika Prasad

1BM19CS081

23.11.2020

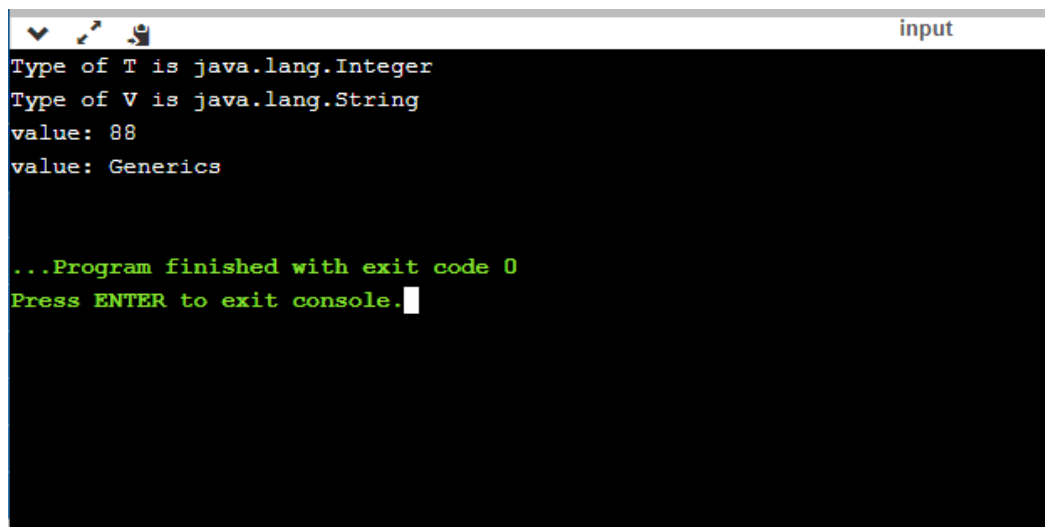
Program 6-

Write a program to demonstrate generics with multiple object parameters.

```
class TwoGen<T, V> {  
    T ob1;  
    V ob2;  
    TwoGen(T o1, V o2) {  
        ob1 = o1;  
        ob2 = o2;  
    }  
    void showTypes() {  
        System.out.println("Type of T is " + ob1.getClass().getName());  
        System.out.println("Type of V is " + ob2.getClass().getName());  
    }  
    T getob1() {  
        return ob1;  
    }  
    V getob2() {  
        return ob2;  
    }  
}
```

```
class Main {  
    public static void main(String args[]) {  
        TwoGen<Integer, String> tgObj = new TwoGen<Integer, String>(88, "Generics");  
        tgObj.showTypes();  
        int v = tgObj.getob1();  
        System.out.println("value: " + v);  
        String str = tgObj.getob2();  
        System.out.println("value: " + str);  
    }  
}
```

Output-

A screenshot of a Java IDE's console window. The window has a title bar with standard OS icons and the text 'input'. The console output is as follows:
Type of T is java.lang.Integer
Type of V is java.lang.String
value: 88
value: Generics

...Program finished with exit code 0
Press ENTER to exit console.
The text is displayed in a monospaced font on a dark background. The first four lines are in white, and the last two lines are in a light green color. A white cursor is visible at the end of the last line.

```
input  
Type of T is java.lang.Integer  
Type of V is java.lang.String  
value: 88  
value: Generics  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Program 8-

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=father’s age.

```
import java.util.Scanner;
```

```
class WrongAge extends Exception {
```

```
    public WrongAge(String s){
```

```
        super(s);
```

```
    }
```

```
}
```

```
class Father {
```

```
    int fatherAge;
```

```
    Father(int fAge, int sAge) throws WrongAge{
```

```
        if(fAge <= sAge){
```

```
            throw new WrongAge("Son's age can't be greater than or equal to father's age");
```

```
        }else{
```

```
            this.fatherAge = fAge;
```

```
        }
```

```
    }
```

```
}
```

```
class Son extends Father {
```

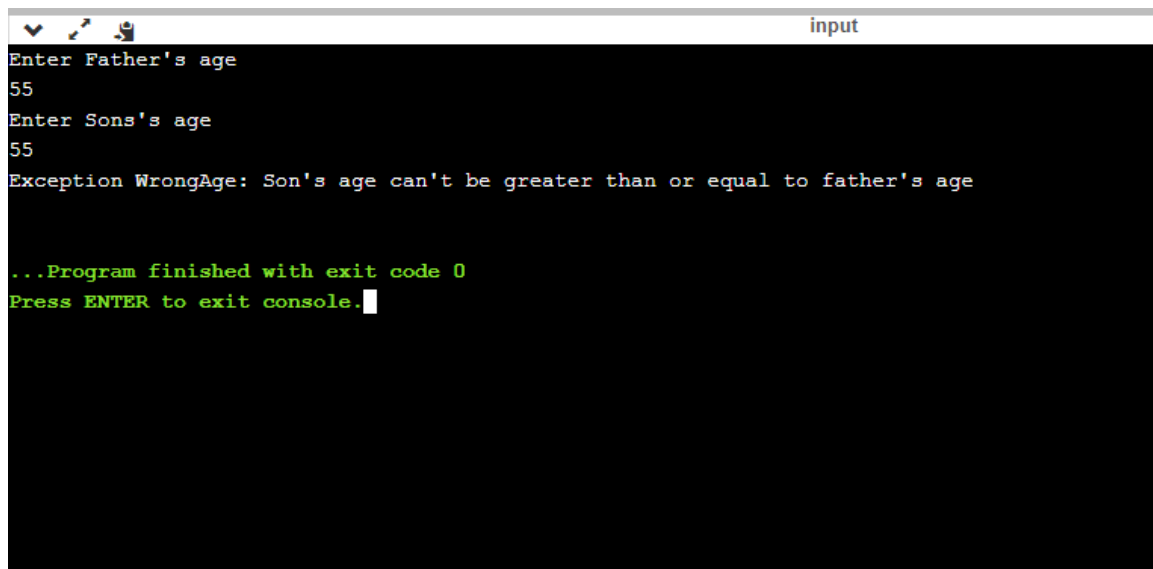
```
    int sonAge;
```

```
Son(int fAge, int sAge) throws WrongAge{  
    super(fAge, sAge);  
    this.sonAge = sAge;  
}  
  
void print(){  
    System.out.println("Father's Age: " + fatherAge);  
    System.out.println("Son's Age: " + sonAge);  
}  
}
```

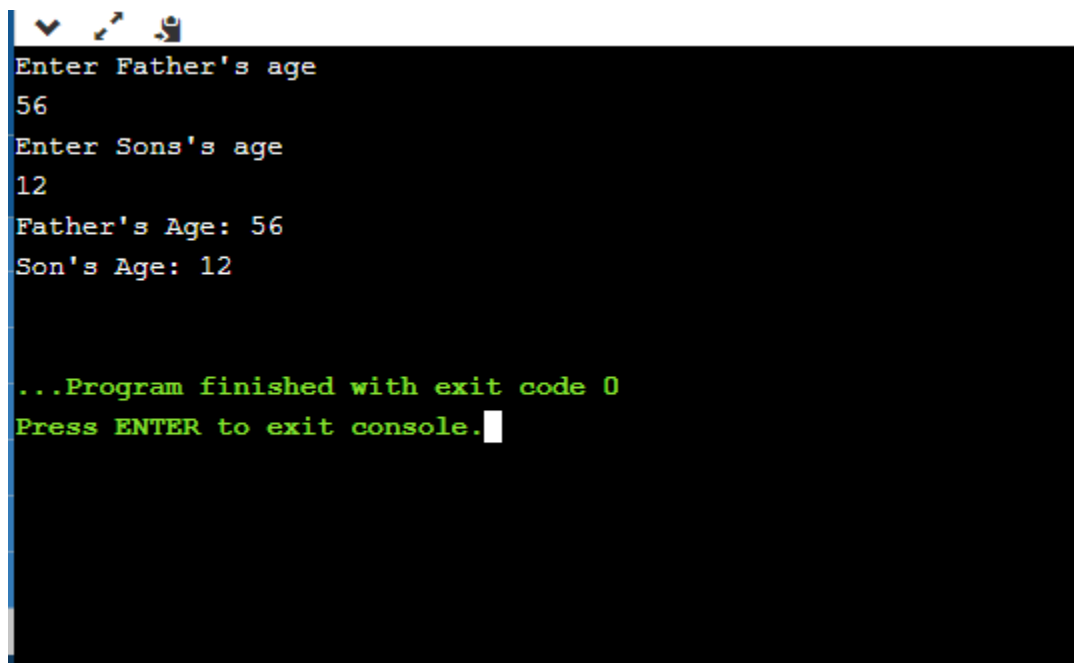
```
public class Main {  
    public static void main(String[] args){  
        int fAge,sAge;  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter Father's age");  
        fAge = sc.nextInt();  
        System.out.println("Enter Sons's age");  
        sAge = sc.nextInt();  
  
        try{  
            Son son = new Son(fAge, sAge);  
            son.print();  
        }  
    }  
}
```

```
}catch(WrongAge err){  
    System.out.println("Exception " + err);  
}  
}  
}
```

Output-



```
Enter Father's age  
55  
Enter Sons's age  
55  
Exception WrongAge: Son's age can't be greater than or equal to father's age  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```



```
Enter Father's age  
56  
Enter Sons's age  
12  
Father's Age: 56  
Son's Age: 12  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```



```
input
Enter Father's age
45
Enter Sons's age
55
Exception WrongAge: Son's age can't be greater than or equal to father's age

...Program finished with exit code 0
Press ENTER to exit console.
```