

Core JAVA training - index

Date:05/08/2024

1. Language And Applications
2. JAVA Features
 - Why is Java Independent?
 - Oops
 - Exception Handling
 - Multithreading
 - Web Application
 - Open Source
 - Security
 - Support Networking
 - Memory Management
3. JDK,JRE,JVM
4. Basic Java Programming
5. Packages

Date:06/08/2024

Mrng(11:00 am)

1. Nested Loops
2. One Dimensional Array
3. Two Dimensional Array
4. Logical Programming

AfterNoon(3:30 pm)

1. SwitchCase
2. Scanner Class
3. Java.lang
 - Object Class Methods
4. Enum
5. Event Management Application

Date:07/08/2024

Mrng(11:00 am)

1.oops

- Encapsulation
- Programs
- Calculation
- Person
- MethodFlow

AfterNoon(3:30 pm)

1. Inheritance
2. Polymorphism
 - .Method overloading
 - .Method Overriding
3. Abstraction
4. IS-A (Inheritance)
5. HAs-A (Object Creation).

Date:08/08/24

Constructor

- i. Class name and constructor name should be same
- ii. There are 2 types of constructors
 - a. Default Constructor
 - b. Parameterized Constructor
- iii. We can access constructor while creation of object
- iv. Constructors are mainly for initializing
- v. Constructor doesn't have any return type not even void. If you declare as a void the compiler will consider as a method not a constructor
- vi. Every class needs atleast 1 default constructor
- vii. this, super This
 - > this is a keyword always refers to instance variables
- viii. Always constructor are overloaded

Program1:

```
1 package com.evergent.corejava.constructor;
2
3 public class Employee1 {
4
5     public Employee1()
6     {
7         System.out.println("Default constructor..");
8     }
9     public static void main(String[] args) {
10         new Employee1();
11     }
12 }
13
```

Problems Declaration Console

<terminated> Employee1 [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\cli

Default constructor..

Program2:

```
1 package com.evergent.corejava.constructor;
2
3 public class Employee2 {
4     int eno;
5     String ename;
6     double sal;
7     public Employee2()
8     {
9         System.out.println("default constructor...");
10    }
11    public Employee2(int eno1, String ename1, double sal1)
12    {
13        eno=enol;
14        ename=ename1;
15        sal=sal1;
16    }
17    public void display()
18    {
19        System.out.println("employee no.:"+eno);
20        System.out.println("employee name.:"+ename);
21        System.out.println("employee sal:"+sal);
22    }
23    public static void main(String[] args) {
24        Employee2 empl=new Employee2();
25        Employee2 emp2=new Employee2(123, "Shiva", 60000);
26        empl.display();
27        emp2.display();
28
29    }
30 }
31
32 }
```

<terminated> Employee2 [Java Application] C

default constructor...
employee no.:0
employee name.:null
employee sal:0.0
employee no.:123
employee name.:Shiva
employee sal:60000.0

Program3:

```

package com.evergent.corejava.constructor;
public class Employee3 {
    int eno;
    String ename;
    double sal;
    public Employee3()
    {
        System.out.println("default constructor...");
    }
    public Employee3(int eno, String ename, double sal)
    {
        this.eno=eno;
        this.ename=ename;
        this.sal=sal;
    }
    public void display()
    {
        System.out.println("employee no.:"+eno);
        System.out.println("employee name.:"+ename);
        System.out.println("employee sal:"+sal);
    }
    public static void main(String[] args) {
        Employee3 emp1=new Employee3();
        Employee3 emp2=new Employee3(123, "Shiva", 60000);
        emp1.display();
        emp2.display();
    }
}

```

```

<terminated> Employee2 Java Application
default constructor...
employee no.:0
employee name.:null
employee sal:0.0
employee no.:123
employee name.:Shiva
employee sal:60000.0

```

Program4:

```

package com.evergent.corejava.constructor;
public class Employee4 {
    void Employee4()
    {
        System.out.println("Default constructor..");
    }
    public static void main(String[] args) {
        Employee4 emp4=new Employee4();
        emp4.Employee4();
    }
}

```

```

<terminated> Employee4 Java Application
Default constructor..

```

Program5:

```

package com.evergent.corejava.constructor;

public class Employee5 {
    int eno;
    String ename;
    double sal;
    public Employee5()
    {
        System.out.println("default constructor... ");
    }
    public Employee5(int eno)
    {
        this.eno=eno;
    }
    public Employee5(int eno, String ename, double sal)
    {
        this.eno=enono;
        this.ename=ename;
        this.sal=sal;
    }
    public void display()
    {
        System.out.println("employee no.:" + eno);
        System.out.println("employee name.:" + ename);
        System.out.println("employee sal.:" + sal);
    }
    public static void main(String[] args) {
        Employee5 emp1=new Employee5();
        Employee5 emp2=new Employee5(123, "Shiva", 60000);
        emp1.display();
        emp2.display();
    }
}

```

```

<terminated> Employee5 [Java Application]
default constructor...
employee no.:0
employee name.:null
employee sal:0.0
employee no.:123
employee name.:Shiva
employee sal:60000.0

```

Program 6:

```

class MyEmployee
{
    int eno;
    public MyEmployee()
    {
    }
    public MyEmployee(int eno) {
        System.out.println("My employee no.:" + eno);
    }
}
public class Employee6 extends MyEmployee{
    String name;
    double sal;
    Employee6()
    {
        System.out.println("default constructor..");
    }
    Employee6(int eno, String name, double sal)
    {
        super(eno);
        this.eno=enono;
        this.name=name;
        this.sal=sal;
    }
    public void display()
    {
        System.out.println("employee no.:" + eno);
        System.out.println("employee name.:" + name);
        System.out.println("employee sal.:" + sal);
    }
    public static void main(String[] args) {
        Employee6 emp1=new Employee6();
        Employee6 emp2=new Employee6(123, "Shiva", 60000);
        emp1.display();
        emp2.display();
    }
}

```

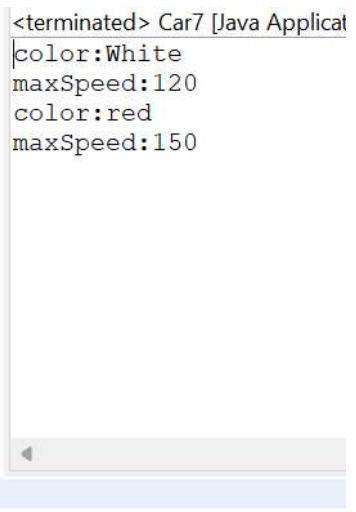
```

<terminated> Employee6 [Java Application]
default constructor..
My employee no.:123
employee no.:0
employee name.:null
employee sal:0.0
employee no.:123
employee name.:Shiva
employee sal:60000.0

```

Program 7:

```
1 package com.evergent.corejava.constructor;
2
3 public class Car7 {
4     String color;
5     int maxSpeed;
6     Car7()
7     {
8         color="White";
9         maxSpeed=120;
10    }
11    Car7(String color,int maxSpeed)
12    {
13        this.color=color;
14        this.maxSpeed=maxSpeed;
15    }
16    void display()
17    {
18        System.out.println("color:"+color);
19        System.out.println("maxSpeed:"+maxSpeed);
20    }
21    public static void main(String[] args) {
22        Car7 c1=new Car7();
23        Car7 c2=new Car7("red",150);
24        c1.display();
25        c2.display();
26    }
27}
28}
```



<terminated> Car7 [Java Application]
color:White
maxSpeed:120
color:red
maxSpeed:150

Program8:

```

class Animal
{
    private String name;
    private int age;
    public Animal(String name,int age)
    {
        this.name=name;
        this.age=age;
    }
    public void displayInfo() {
        System.out.println("Name:"+name);
        System.out.println("Age:"+age);
    }
}
class Dog extends Animal{
    private String breed;
    public Dog(String name,int age,String breed)
    {
        super(name,age);
        this.breed=breed;
    }
    public void displayInfo()
    {
        super.displayInfo();
        System.out.println("Breed:"+breed);
    }
}
public class Inheritance_OVERRIDING8 {
    public static void main(String[] args) {
        Dog dog=new Dog("Buddy",5,"Golden Retriever");
        dog.displayInfo();
    }
}

```

```

<terminated> Inheritance_OVERRIDING8
Name:Buddy
Age:5
Breed:Golden Retriever

```

Program9:

```
package com.evergreen.corejava.construction;

public class Student9 {
    String name;
    int age;
    public Student9(String name,int age)
    {
        this.name=name;
        this.age=age;
    }
    public Student9(Student9 s)
    {
        this.name=s.name;
        this.age=s.age;
    }
    public void displayDetails()
    {
        System.out.println("Name:"+name);
        System.out.println("Age:"+age);
    }
    public static void main(String[] args) {
        Student9 student1=new Student9("shiva",20);
        Student9 student2=new Student9(student1);
        student1.displayDetails();
        student2.displayDetails();
    }
}
```

```
<terminated> Student9 java
Name:shiva
Age:20
Name:shiva
Age:20
```

Date:09/08/2024 - Day5

1. Static

- a. Static is a keyword
- b. We can declare variables and methods as static
- c. We can access static variables and static methods directly through `classname.methodname` and `classname.variablename` respectively.
- d. Static methods can access static methods and static variables only.
- e. Static methods cannot access non static methods and non static variables.
- f. Non static methods can access static methods and static variables.
- g. Static block- whenever class is loaded inside the JVM at that time static block is initiated.

Program1:

```
1 package com.evergent.corejava.staticprograms;
2
3 public class StaticDemo1 {
4     static String cname="India";
5     static public void myData()
6     {
7         System.out.println("static method:myData");
8     }
9     public static void main(String[] args) {
0         System.out.println(cname);
1         myData();
2     }
3 }
```

```
<terminated> StaticDemo1 [Java Application] C:\Use
|India
|static method:myData
```

Program2:

```
1 package com.evergent.corejava.staticprograms;
2
3 public class StaticDemo2 {
4     static String cname="India";
5     static public void myData()
6     {
7         System.out.println("static method:myData");
8     }
9     public static void main(String[] args) {
0         System.out.println(StaticDemo2.cname);
1         StaticDemo2.myData();
2     }
3 }
```

```
<terminated> StaticDemo2 [Java
|India
|static method:myData
```

Program3:

```
package com.evergent.corejava.staticprograms;

public class StaticDemo3 {
    static String cname="India";
    String name="shiva";
    static public void myData()
    {
        System.out.println("my data..");
        //myShow(); static method can't access non static method
    }
    public void myShow()
    {
        System.out.println("my show");
    }
    public static void main(String[] args) {
        //name;           Cannot make a static reference to the non-static field name
        myData();
    }
}
```

<terminated> StaticDemo3 [Java]
my data..

Program4:

```
package com.evergent.corejava.staticprograms;

public class StaticDemo4 {
    static String cName="India";
    String name="Shiva";
    static public void myData()
    {
        System.out.println("my data:"+cName);
    }
    public void myShow()
    {
        myData();
        System.out.println(cName);
    }
    public static void main(String[] args) {
        myData();
        System.out.println(cName);
        StaticDemo4 s4=new StaticDemo4();
        s4.myShow();
    }
}
```

<terminated> StaticDemo4 [Java]
my data:India
India
my data:India
India

Program5:

```

package com.evergent.corejava.staticprograms;

public class StaticDemo5 {
    static
    {
        System.out.println("static block:open db");
    }
    static String cname="India";
    static public void myData()
    {
        System.out.println("my data");
    }
    public static void main(String[] args) {
        System.out.println(StaticDemo5.cname);
        StaticDemo5.myData();
    }
}

```

```

<terminated> StaticDemo5 [Java Application] C:\Users\Shivani.Ji
static block:open db/network connection
India
my data

```

Program6:

```

package com.evergent.corejava.staticprograms;
//if we modify static variable it reflected glo
public class Person6 {
    static String name="shiva";
    int age=22;
    String address="Hyderabad";
    public void display()
    {
        name="welcome";
        System.out.println("Name:"+name);
        System.out.println("Age:"+age);
        System.out.println("Address:"+address);
    }
    public static void main(String[] args) {
        Person6 p1=new Person6();
        //System.out.println(name);
        p1.display();
        //System.out.println(name);
        Person6 p2=new Person6();
        System.out.println(name);
    }
}

```

```

<terminated> Person6 [Java , 
Name:welcome
Age:22
Address:Hyderabad
welcome

```

2. Final

- a. Final is a Keyword.
- b. We can declare a variable, method, or a class as final.
- c. Final variable cannot be modified.
- d. Final Method cannot be overrided.
- e. Final class cannot be inherited.

Program1:

```
package com.evergent.corejava.finalprograms;

public class FinalDemo1 {
    final String cname="India";
    public void myData()
    {
        //cname="welcome"; final cname  cannot
        System.out.println(cname);
    }
    public static void main(String[] args) {
        FinalDemo1 fd1=new FinalDemo1();
        fd1.myData();
    }
}
```

```
<terminated> FinalDemo1
India
```

Program2:

```
package com.evergent.corejava.finalprograms;

class MyClass{
    final public void myProducts()
    {
        System.out.println("AI products..");
    }
}

public class FinalDemo2 extends MyClass {

    final String cname="India";
    /*Cannot override the final method from MyClass*/
    public void myProducts()
    {
        System.out.println("AI products..");
    }*/
    public void myData()
    {
        System.out.println(cname);
    }
    public static void main(String[] args) {
        FinalDemo2 fd1=new FinalDemo2();
        fd1.myData();
    }
}
```

Program3:

```

package com.evergent.corejava.finalprograms;

final class MyClass1{
    final public void myProducts()
    {
        System.out.println("AI products..");
    }
}
//The type FinalDemo3 cannot subclass the final
public class FinalDemo3  {

    final String cname="India";
    /*Cannot override the final method from MyClass1*/
    public void myProducts()
    {
        System.out.println("AI products..");
    }*/
    public void myData()
    {
        System.out.println(cname);
    }
    public static void main(String[] args) {
        FinalDemo3 fd1=new FinalDemo3();
        fd1.myData();
        MyClass1 ml=new MyClass1();
        ml.myProducts();

    }
}

```

```

<terminated> FinalDemo3
|India
AI products..

```

Date:12/08/2024 - Day6

1. String

- a. why String is immutable?
- b. String is final class
- c. String class having methods.
- d. All string class methods are non-synchronized.

METHODS:

1. length();

2. toLowercase();
3. toUpperCase();

2. StringBuffer

- a. StringBuffer is final class
- b. StringBuffer is mutable
- c. StringBuffer having methods
- d. All StringBuffer class methods are synchronized.

METHODS:

1. append();
2. insert();
3. replace();
4. delete();
5. reverse();
6. capacity();
7. length();

3. StringBuilder

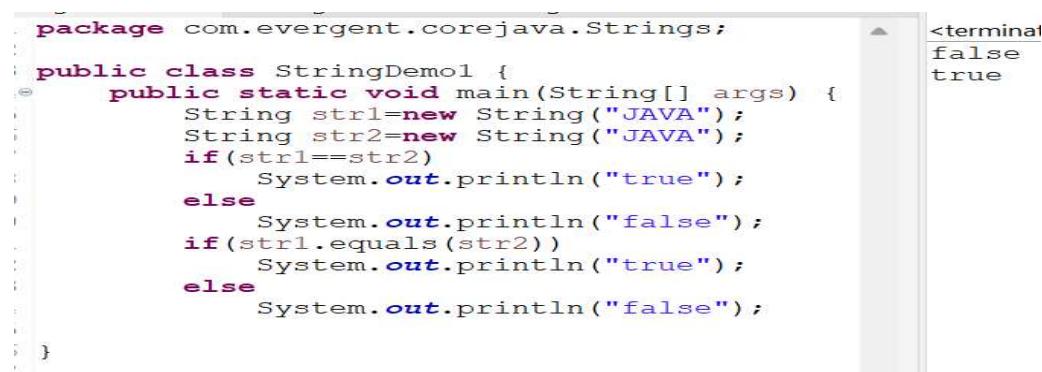
- a. StringBuilder is final class
- b. StringBuilder is mutable
- c. StringBuilder having methods
- d. All StringBuilder class methods are non-synchronized.

METHODS:

1. append();
2. insert();
3. replace();
4. delete();
5. reverse();

4. Difference between String, StringBuffer, StringBuilder

Program1:



The image shows a Java code editor and a terminal window. The code in the editor is as follows:

```
package com.evergent.corejava.Strings;
public class StringDemo1 {
    public static void main(String[] args) {
        String str1=new String("JAVA");
        String str2=new String("JAVA");
        if(str1==str2)
            System.out.println("true");
        else
            System.out.println("false");
        if(str1.equals(str2))
            System.out.println("true");
        else
            System.out.println("false");
    }
}
```

The terminal window to the right shows the output of the program:

```
<terminal>
false
true
```

Program2:

```
public class StringDemo2 {  
    public static void main(String[] args) {  
        String s1="JAVA";  
        String s2="JAVA";  
        if(s1==s2)  
            System.out.println("True");  
        else  
            System.out.println("False");  
        if(s1.equals(s2))  
            System.out.println("true");  
        else  
            System.out.println("false");  
    }  
}
```

True
true

Program3:

```
public class StringDemo3_Methods {  
    public static void main(String[] args) {  
        String name=new String(" Shivani ")  
        System.out.println(name.length());  
        System.out.println(name.toUpperCase());  
        System.out.println(name.toLowerCase());  
        System.out.println(name.trim());  
    }  
}
```

11
SHIVANI
shivani
Shivani

Program4:

```
package com.evergent.corejava.strings;  
  
public class SubString {  
    public static void main(String[] args) {  
        String str="the Quick brown fox jumps  
        String subStr="fox";  
        boolean contains=str.contains(subStr)  
        System.out.println("contains '" + subSt  
    }  
}
```

<terminated> SubString
contains 'fox':true

Program5:

```
public class RemoveSpace {  
    public static void main(String[] args)  
        String str="Hello World, this is a tes  
        String noSpaces=str.replace(" ","");  
        System.out.println(noSpaces);  
    }  
}
```

HelloWorld, this is a test

Program6:

```
public class StringConcat {  
    public static void main(String[] args) {  
        String str="Hello";  
        str=str.concat(" World!");  
        System.out.println(str);  
    }  
}
```

Hello World

Program7:

```
1 package com.evergent.corejava.Strings;
2
3 public class ReverseString {
4     public static void main(String[] args) {
5         String str="Hello World!";
6         StringBuilder reversed=new StringBuilder(str);
7         System.out.println(reversed);
8     }
9 }
0 }
```

```
<terminated> ReverseString
!dlrow olleH
```

Program8:

```
package com.evergent.corejava.Strings;
public class SplitDemo1 {
    public static void main(String[] args) {
        String str="Java is a powerful program";
        String[] words=str.split(" ");
        for(int i=0;i<words.length;i++)
        {
            System.out.println(words[i]);
        }
    }
}
```

```
<terminated> SplitDemo1
Java
is
a
powerful
programming
language
```

Program9:

```
public class SplitDemo2 {
    public static void main(String[] args) {
        String str="Java is a powerful program";
        String[] words=str.split(" ");
        for(String s:words)
        {
            System.out.println(s);
        }
    }
}
```

```
Java
is
a
powerful
programming
language
```

Program10:

```
public class StringBufferExample {
    public static void main(String[] args) {
        StringBuffer sb=new StringBuffer("Hello");
        System.out.println("initial String:"+sb);
        //append a string
        sb.append(" world!");
        System.out.println("After append: "+sb);
        //Insert a string at a specific position
        sb.insert(6, "beautiful");
        System.out.println("After insert: "+sb);
        //replace a substring
        sb.replace(0, 4, "hi");
        System.out.println("After replace: "+sb);
        //delete a substring
        sb.delete(0, 3);
        System.out.println("After delete: "+sb);
        //reverse the string
        sb.reverse();
        System.out.println("after reverse :"+sb);
        //capacity and length
        System.out.println("capacity:"+sb.capacity());
        System.out.println("length:"+sb.length());
    }
}
```

```
initial String:Hello
After append: Hello world!
After insert:Hello beautifulworld!
After replace:hio beautifulworld!
After delete: beautifulworld!
after reverse :!dlrowlufituaeb
capacity:21
length:16
```

Program11:

```
package com.evergent.corejava.Strings;

public class StringBuilderExample {
    public static void main(String[] args) {
        StringBuilder sb=new StringBuilder("Hello");
        System.out.println("initial String:"+sb);
        //append a string
        sb.append(" world!");
        System.out.println("After append: "+sb);
        //Insert a string at a specific position
        sb.insert(6, "beautiful");
        System.out.println("After insert:"+sb);
        //replace a substring
        sb.replace(0, 4, "hi");
        System.out.println("After replace:"+sb);
        //delete a substring
        sb.delete(0, 3);
        System.out.println("After delete:"+sb);
        //reverse the string
        sb.reverse();
        System.out.println("after reverse :" +sb);
    }
}
```

```
<terminated> StringBuilderExample java Application
initial String:Hello
After append: Hello world!
After insert:Hello beautifulworld!
After replace:hio beautifulworld!
After delete: beautifulworld!
after reverse :!dlrowlufituaeb
```

Program12:

```
public class StringPerformance1 {
    public static void main(String[] args) {
        String a;
        String b;
        System.out.println("a"+b);
        System.out.println((char) ('a'+3));
    }
}
```

```
term
ab
d
```

Program13:

```
public class StringPerformance2 {
    public static void main(String[] args) {
        String a;
        String b;
        System.out.println('a'+'b');
        System.out.println('a'+3);
    }
}
```

```
term
195
100
```

Program14:

```
package com.evergent.corejava.Strings;

public class StringPerformance3 {
    public static void main(String[] args) {
        StringBuilder builder=new StringBuilder();
        for(int i=0;i<26;i++)
        {
            char ch=(char) ('a'+i);
            builder.append(ch);
        }
        System.out.println(builder);
    }
}
```

```
<terminated> StringPerformance3 [Java A
abcdefghijklmnopqrstuvwxyz
```

Program15:

```
package com.evergent.corejava.Strings;

public class StringPerformance4 {
    public static void main(String[] args) {
        String series="";
        for(int i=0;i<26;i++)
        {
            char ch=(char) ('a'+i);
            series=series+ch;
        }
        System.out.println(series);
    }
}
```

```
<terminated> StringPerformance4 [Java]
abcdefghijklmnopqrstuvwxyz
```

Program16:

```
package com.evergent.corejava.Strings;

import java.util.Arrays;

public class StringPerformance5 {
    public static void main(String[] args) {
        String name="JavaTechnologies";
        System.out.println(Arrays.toString(name.toCharArray()));
        System.out.println(name.indexOf('T'));
        System.out.println(" JAVA ".strip());
    }
}
```

```
<terminated> StringPerformance5 [Java Application] C:\Users\Shivani.Jakkula
[J, a, v, a, T, e, c, h, n, o, l, o, g, i, e, s]
4
JAVA
```

Date:13/08/2024 - Day7

- **Can we make class as immutable?**

Yes, by declaring the class as final, as well as declaring the variable as private and final.

- Object class method - `toString()`
- String immutable
- Interface
 - 1 Interface is a keyword.
 - 2 we can declare method signature only but not implementation.
 - 3 By default all interface methods are abstract.
 - 4 If any class implements interface the class should be override all interface methods otherwise the class will be showing compile time error.
 - 5 we cannot create object to interface but we can create reference to interface.
 - 6. We can declare variables inside interface all are public static final.
 - 7. Java will support multiple inheritance through interface.
 - 8. One class can implements more than one interface.
 - 9. One interface can extend other interface.

Program1: MainPerson, PersonImmutable

```
1 package com.evergent.corejava.string.immutable;
2
3 public final class PersonImmutable {
4     private final String name;
5     private final int age;
6     public PersonImmutable(String name,int age)
7     {
8         this.name=name;
9         this.age=age;
10    }
11    public String myName()
12    {
13        return name;
14    }
15    public int myAge()
16    {
17        return age;
18    }
19 }
```

```
1 package com.evergent.corejava.string.immutable;
2
3 public class MainPerson {
4     public static void main(String[] args) {
5         PersonImmutable person=new PersonImmutable("Shiva",30);
6         System.out.println("Name: "+person.myName());
7         System.out.println("Age: "+person.myAge());
8     }
9 }
10
11
```

Problems Declaration Console

terminated> MainPerson [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v202

Name: Shiva

Age: 30

Program2:

```
1 package com.evergent.corejava.objectclassmethods;
2
3 class Person{
4     String name;
5     int age;
6     public Person(String name, int age) {
7         this.name = name;
8         this.age = age;
9     }
10    public String toString()
11    {
12        return "name: "+name+" age: "+age;
13    }
14 }
15
16 public class MyPerson {
17     public static void main(String[] args) {
18         Person p=new Person("shiva",22);
19         System.out.println(p);
20         System.out.println(p.hashCode());
21     }
22 }
```

terminated> MyPerson [Java Application]

name: shiva age: 22

212628335

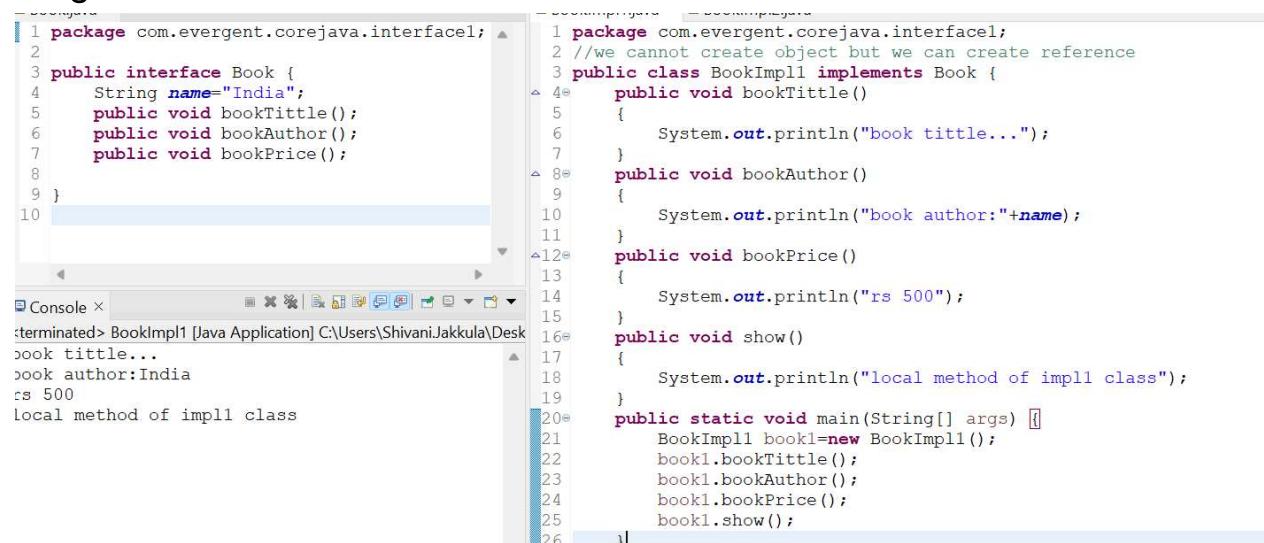
Program3:

```
1 package com.evergent.corejava.string.immutable;
2
3 final class ImmutableString{
4     private final String value;
5
6     public ImmutableString(String value) {
7         this.value = value;
8     }
9     public String toString()
10    {
11        return value;
12    }
13 }
14
15 public class MyData {
16     public static void main(String[] args) {
17         ImmutableString str=new ImmutableString("Hello String world...");
18         System.out.println(str.toString());
19     }
20 }
```

terminated> MyData [Java Application]

hello String world...

Program4:

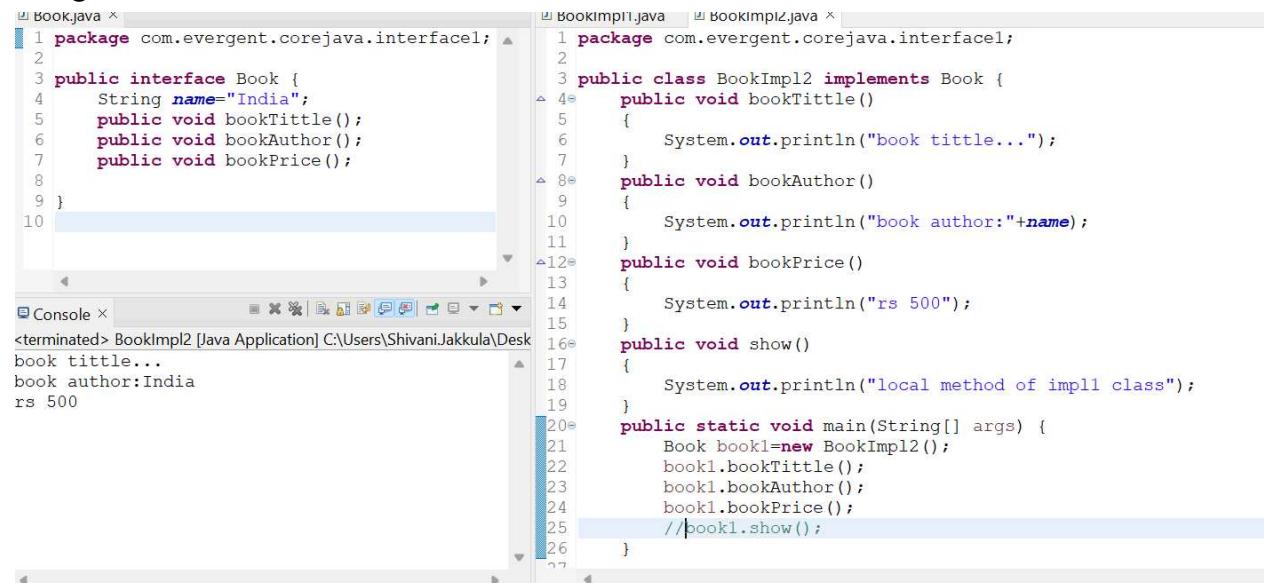


```
1 package com.evergent.corejava.interface1;
2
3 public interface Book {
4     String name="India";
5     public void bookTittle();
6     public void bookAuthor();
7     public void bookPrice();
8 }
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
```

```
1 package com.evergent.corejava.interface1;
2 //we cannot create object but we can create reference
3 public class BookImpl1 implements Book {
4     public void bookTittle()
5     {
6         System.out.println("book tittle...");
7     }
8     public void bookAuthor()
9     {
10        System.out.println("book author:"+name);
11    }
12     public void bookPrice()
13    {
14        System.out.println("rs 500");
15    }
16     public void show()
17    {
18        System.out.println("local method of impl1 class");
19    }
20     public static void main(String[] args) {
21         BookImpl1 book1=new BookImpl1();
22         book1.bookTittle();
23         book1.bookAuthor();
24         book1.bookPrice();
25         book1.show();
26     }
27 }
```

```
Console x
terminated> BookImpl1 [Java Application] C:\Users\ShivaniJakkula\Desktop
book tittle...
book author:India
rs 500
local method of impl1 class
```

Program5:

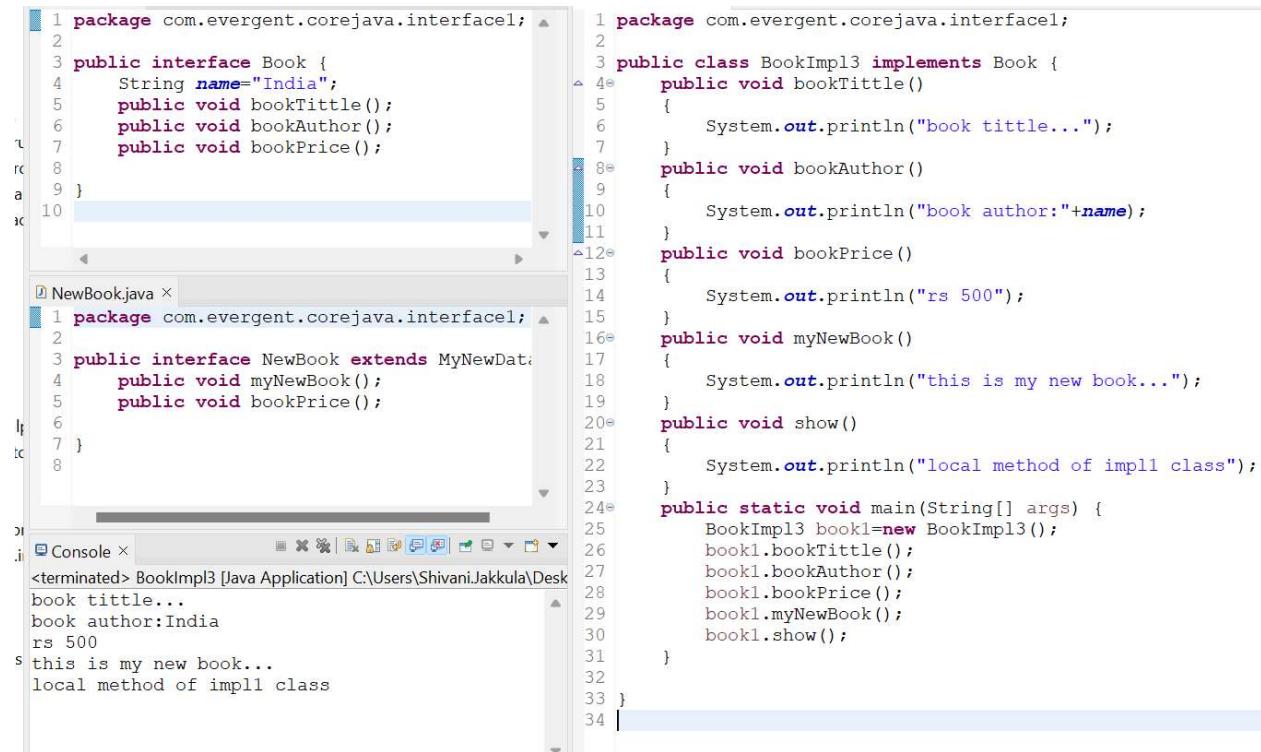


```
1 package com.evergent.corejava.interface1;
2
3 public interface Book {
4     String name="India";
5     public void bookTittle();
6     public void bookAuthor();
7     public void bookPrice();
8 }
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
```

```
1 package com.evergent.corejava.interface1;
2
3 public class BookImpl2 implements Book {
4     public void bookTittle()
5     {
6         System.out.println("book tittle...");
7     }
8     public void bookAuthor()
9     {
10        System.out.println("book author:"+name);
11    }
12     public void bookPrice()
13    {
14        System.out.println("rs 500");
15    }
16     public void show()
17    {
18        System.out.println("local method of impl1 class");
19    }
20     public static void main(String[] args) {
21         Book book1=new BookImpl2();
22         book1.bookTittle();
23         book1.bookAuthor();
24         book1.bookPrice();
25         //book1.show();
26     }
27 }
```

```
Console x
terminated> BookImpl2 [Java Application] C:\Users\ShivaniJakkula\Desktop
book tittle...
book author:India
rs 500
local method of impl1 class
local method of impl2 class
```

Program6:



```
1 package com.evergent.corejava.interface1;
2
3 public interface Book {
4     String name="India";
5     public void bookTitle();
6     public void bookAuthor();
7     public void bookPrice();
8 }
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2179
2180
2181
2182
2183
2184
2185

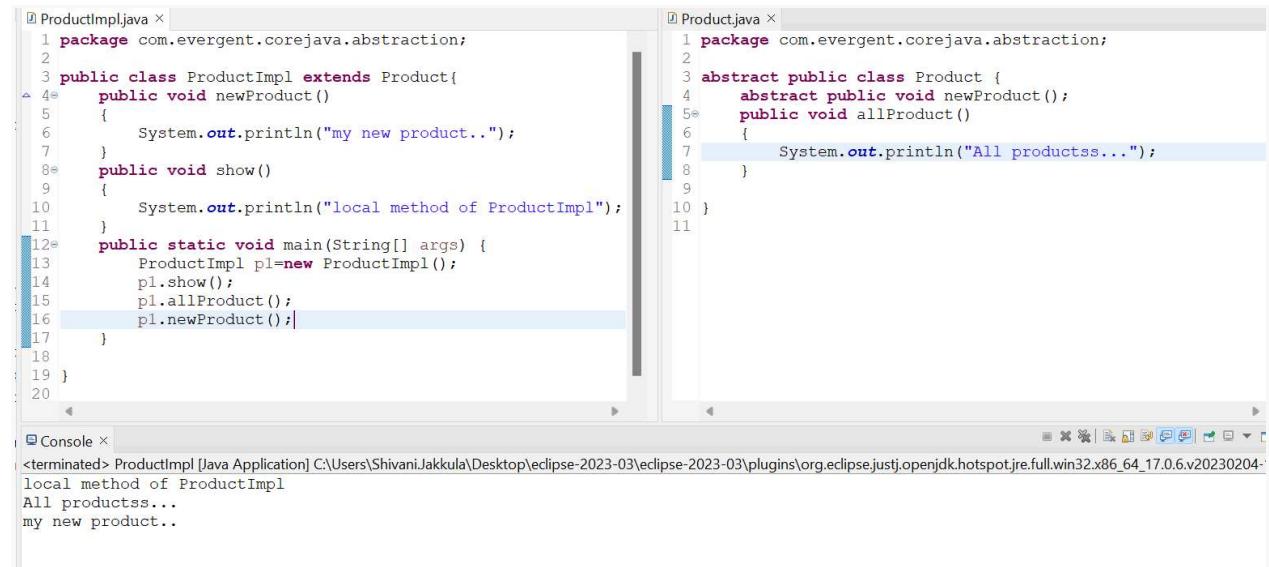
```

Date:14/08/2024 - Day8

● Abstract class

1. Abstract is a keyword
2. Abstract class having abstract and concrete methods(implemented).
3. If any class having one abstract method that class should be declare as abstract class ,otherwise the abstract class should show compile time error.
4. If any class extends abstract class that class should be override all abstract methods otherwise the class will show compile time error.
5. We cannot create object to abstract class but we can create reference to abstract class.
6. Without any abstract methods we can also declare as abstract for the security & we can't create object.

Program1:

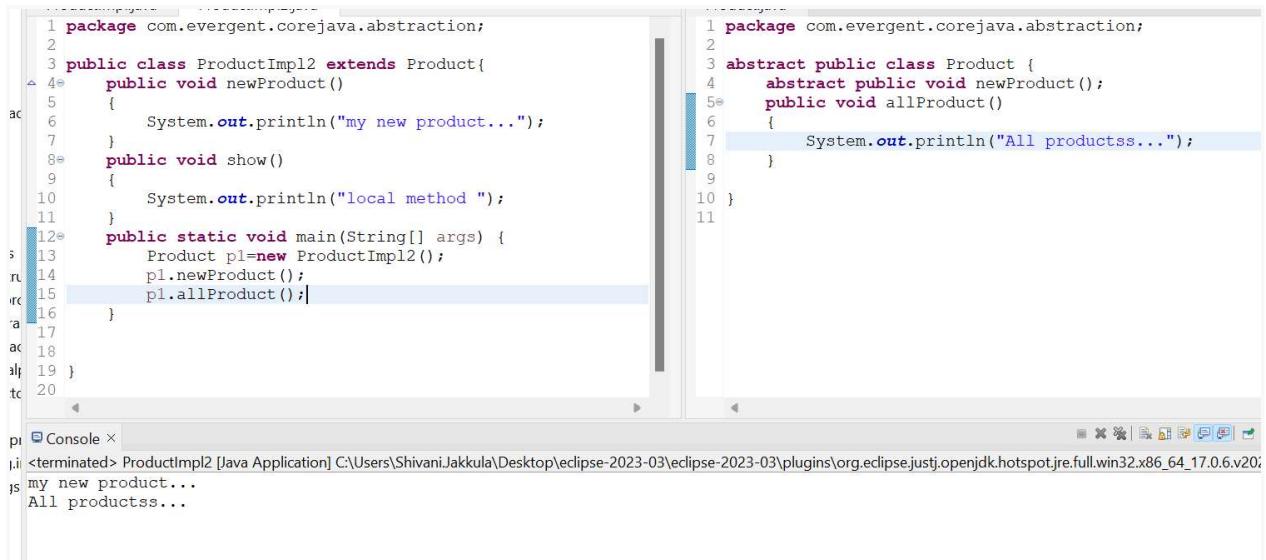


```
ProductImpl.java x
1 package com.evergent.corejava.abstraction;
2
3 public class ProductImpl extends Product{
4     public void newProduct()
5     {
6         System.out.println("my new product..");
7     }
8     public void show()
9     {
10        System.out.println("local method of ProductImpl");
11    }
12    public static void main(String[] args) {
13        ProductImpl pl=new ProductImpl();
14        pl.show();
15        pl.allProduct();
16        pl.newProduct();
17    }
18
19 }
```

```
Product.java x
1 package com.evergent.corejava.abstraction;
2
3 abstract public class Product {
4     abstract public void newProduct();
5     public void allProduct()
6     {
7         System.out.println("All productss...");
8     }
9
10 }
11
```

Console x
<terminated> ProductImpl [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-
local method of ProductImpl
All productss...
my new product..

Program2:



```

1 package com.evergent.corejava.abstraction;
2
3 public class ProductImpl2 extends Product{
4     public void newProduct()
5     {
6         System.out.println("my new product...");
7     }
8     public void show()
9     {
10        System.out.println("local method ");
11    }
12    public static void main(String[] args) {
13        Product p1=new ProductImpl2();
14        p1.newProduct();
15        p1.show();
16    }
17
18}
19
20

```

```

1 package com.evergent.corejava.abstraction;
2
3 abstract public class Product {
4     abstract public void newProduct();
5     public void allProduct()
6     {
7         System.out.println("All productss...");
8     }
9
10 }
11
12

```

Console <terminated> ProductImpl2 [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-19:53:00
my new product...
All productss...

Program3:



```

1 package com.evergent.corejava.abstraction;
2
3 abstract public class Product3 {
4     String cname;
5     public Product3(String cname)
6     {
7         this.cname=cname;
8         System.out.println("called aash");
9     }
10    abstract public void newProduct();
11    public void allProducts()
12    {
13        System.out.println("All products.."+cname);
14    }
15
16}
17
18
19

```

```

1 package com.evergent.corejava.abstraction;
2
3 public class ProductImpl3 extends Product3{
4     public ProductImpl3()
5     {
6         super("Shivani");
7         System.out.println("productimpl3 sub class constructor");
8     }
9
10    public void newProduct()
11    {
12        System.out.println("my new product..");
13    }
14    public void show()
15    {
16        System.out.println("local method");
17    }
18    public static void main(String[] args) {
19        Product3 p1=new ProductImpl3();
20        p1.newProduct();
21    }
22
23
24

```

Console <terminated> ProductImpl3 [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-19:53:00
called aash
productimpl3 sub class constructor
my new product..
All products..Shivani

19/08/2024- Day 10

1. Exception Handling is mechanism.
2. Exceptions are inbuilt mechanism of java.
3. All Exceptions are executed while abnormal conditions only.
4. Normal flow it won't execute any exceptions.
5. Once any exceptions are occurring in java then remaining lines of code is unreachable.
6. Java.lang.throwable is the super class for exception and errors.

7. There are two types of exceptions in Java
 01. Checked exceptions.
 02. Unchecked exceptions.
8. All checked exceptions are compile time exceptions.
9. All unchecked exceptions are runtime exceptions.
10. There are 5 keywords in exception handling
 - a. try{}
 - b. catch(){}
 - c. finally{}
 - d. throws
 - e. throw
11. try is for business logic.
12. catch is for handling exceptions.
13. finally is a block , if exceptions is occurred or not finally block will be execute.
14. throws an exception will be executed method by method.
15. throw is for runtime exceptions and will call predefined exceptions or user defined exceptions.
16. try followed by either catch block or finally block.
17. We should follow exceptions hierarchical.
18. We can create our own (user defined) exceptions.
19. Our own exceptions extends exceptions or runtime exceptions.
20. All exceptions classes are into java.lang package.
21. There is two exceptions in class, Developer should be handle one after one.
22. Developer cannot handled error.

Program 1:

```

1 package com.evergent.corejava.exceptionhandling;
2 /* 1.Exception Handling is mechanism
3  * 2.Exceptions are inbuilt mechanism of java
4 */
5 public class ExceptionDemo1 {
6   public static void main(String[] args) {
7     String name=null;
8     System.out.println(name.length());
9   }
10
11 }

```

Console X

terminated> ExceptionDemo1 [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" because "name" is null
at com.evergent.corejava.exceptionhandling.ExceptionDemo1.main(ExceptionDemo1.java:8)

Program 2:

```
1 package com.evergent.corejava.exceptionhandling;
2 /*3. All Exceptions are executed while abnormal conditions only
3 * 4. Normal flow it won't execute any exceptions.
4 * 5.once any exceptions are occurring in java then remaining lines of code is unreachable.
5 */
6 public class ExceptionDemo2 {
7     String name="null";
8     public void myData()
9     {
10         try {
11             System.out.println("One");
12             System.out.println(name.length());
13             System.out.println("End");
14         }
15         catch(NullPointerException e)
16         {
17             System.out.println("I can handle "+e);
18         }
19     }
20     public static void main(String[] args) {
21         ExceptionDemo2 ed2=new ExceptionDemo2();
22         ed2.myData();
23     }
24 }
25 }
```

Console <terminated> ExceptionDemo2 [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.orient\src\main\java\com\evergent\corejava\exceptionhandling\ExceptionDemo2.java
One
4
End

Program 3:

```
package com.evergent.corejava.exceptionhandling;

public class ExceptionDemo3 {
    String name=null;
    int k=2;
    public void myData()
    {
        try {
            System.out.println("One");
            System.out.println(name.length());
            int t=10/k;
            System.out.println(t);
            System.out.println("End");
        }
        catch(NullPointerException e)
        {
            System.out.println("i could handle"+e);
        }
        catch(ArithmaticException e)
        {
            System.out.println("i could handle:"+e);
        }
    }
    public static void main(String[] args) {
        ExceptionDemo3 ed3=new ExceptionDemo3();
        ed3.myData();
    }
}
```

<terminated> ExceptionDemo3 [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.orient\src\main\java\com\evergent\corejava\exceptionhandling\ExceptionDemo3.java
One
i could handlejava.lang.NullPointerException: Cannot convert null to int at com.evergent.corejava.exceptionhandling.ExceptionDemo3.myData(ExceptionDemo3.java:15)
 at com.evergent.corejava.exceptionhandling.ExceptionDemo3.main(ExceptionDemo3.java:23)

Program 4:

```
package com.evergent.corejava.exceptionhandling;

public class ExceptionDemo4 {
    String name=null;
    int k=0;
    public void myData()
    {
        try {
            System.out.println("one");
            System.out.println(name.length());
            System.out.println(10/k);
            System.out.println("end..");
        }
        catch(NullPointerException e)
        {
            System.out.println("i can handle:"+e);
        }
        catch(ArithmaticException e)
        {
            System.out.println("i can handle:"+e);
        }
        catch(Exception e)
        {
            System.out.println("i can handle:"+e);
        }
    }
    public static void main(String[] args) {
        ExceptionDemo4 emp=new ExceptionDemo4();
        emp.myData();
    }
}
```

```
<terminated> ExceptionDemo4 [Java Application] C:\Users\ShivaniJakkula\Desktop
one
i can handle:java.lang.NullPointerException: Ca

```

Program5:

```
package com.evergent.corejava.exceptionhandling;

public class ExceptionDemo5 {
    String name="null";
    int k=0;
    public void myData()
    {
        try {
            System.out.println("one");
            System.out.println(name.length());
            System.out.println(10/k);
            System.out.println("end..");
        }
        catch(NullPointerException e)
        {
            System.out.println("i can handle:"+e);
        }
        catch(ArithmaticException e)
        {
            System.out.println("i can handle:"+e);
        }
        catch(Exception e)
        {
            System.out.println("i can handle:"+e);
        }
        finally {
            System.out.println("finally block....");
        }
    }
    public static void main(String[] args) {
        ExceptionDemo5 emp=new ExceptionDemo5();
        emp.myData();
    }
}
```

```
<terminated> ExceptionDemo5 [Java Application] C:\Users\ShivaniJakkula\Desktop
one
4
i can handle:java.lang.ArithmaticException: / by zero
finally block....
```

Program 6:

```
package com.evergent.corejava.exceptionhandling;

public class ExceptionDemo6 {
    String name="null";
    int k=2;
    public void myData()
    {
        try {
            System.out.println("one");
            System.out.println(name.length());
            System.out.println(10/k);
            System.out.println("end..");
        }

        finally {
            System.out.println("finally block....");
        }
    }
    public static void main(String[] args) {
        ExceptionDemo6 emp=new ExceptionDemo6();
        emp.myData();
    }
}
```

<terminated> ExceptionDemo6 []
one
4
5
end..
finally block....

20/08/2024- Day 11

Program 7:

```
package com.evergent.corejava.exceptionhandling;

public class ExceptionDemo7Throws {
    String name=null;
    int k=2;
    public void myData() throws NullPointerException{
        System.out.println("One");
        System.out.println(name.length());
        System.out.println(10/k);
        System.out.println("end");
    }
    public static void main(String[] args)
    {
        try {
            ExceptionDemo7Throws ed=new ExceptionDemo7Throws();
            ed.myData();
        }
        catch(Exception e)
        {
            System.out.println("i can handle:"+e);
        }
    }
}
```

<terminated> ExceptionDemo7Throws [Java Application] C:\Users\Shiv
One
i can handle:java.lang.NullPointerException: C:

Program 8:

```
package com.evergent.corejava.exceptionhandling;

public class ExceptionDemo7Throws2 {
    String name=null;
    int k=2;
    public void myData() throws NullPointerException{
        System.out.println("One");
        System.out.println(10/k);
        System.out.println("end");
    }
    public void myChange() throws NullPointerException
    {
        System.out.println("myChange method...");
        System.out.println(name.length());
        myData();
    }
}

public static void main(String[] args)
{
    try {
        ExceptionDemo7Throws2 ed=new ExceptionDemo7Throws2()
        ed.myChange();
    }
    catch(Exception e)
    {
        System.out.println("i can handle:"+e);
    }
}

}
```

```
<terminated> ExceptionDemo7Throws2 [Java Application] C:\Users\Shivani
myChange method...
i can handle:java.lang.NullPointerException: Cann
```

Program 9:

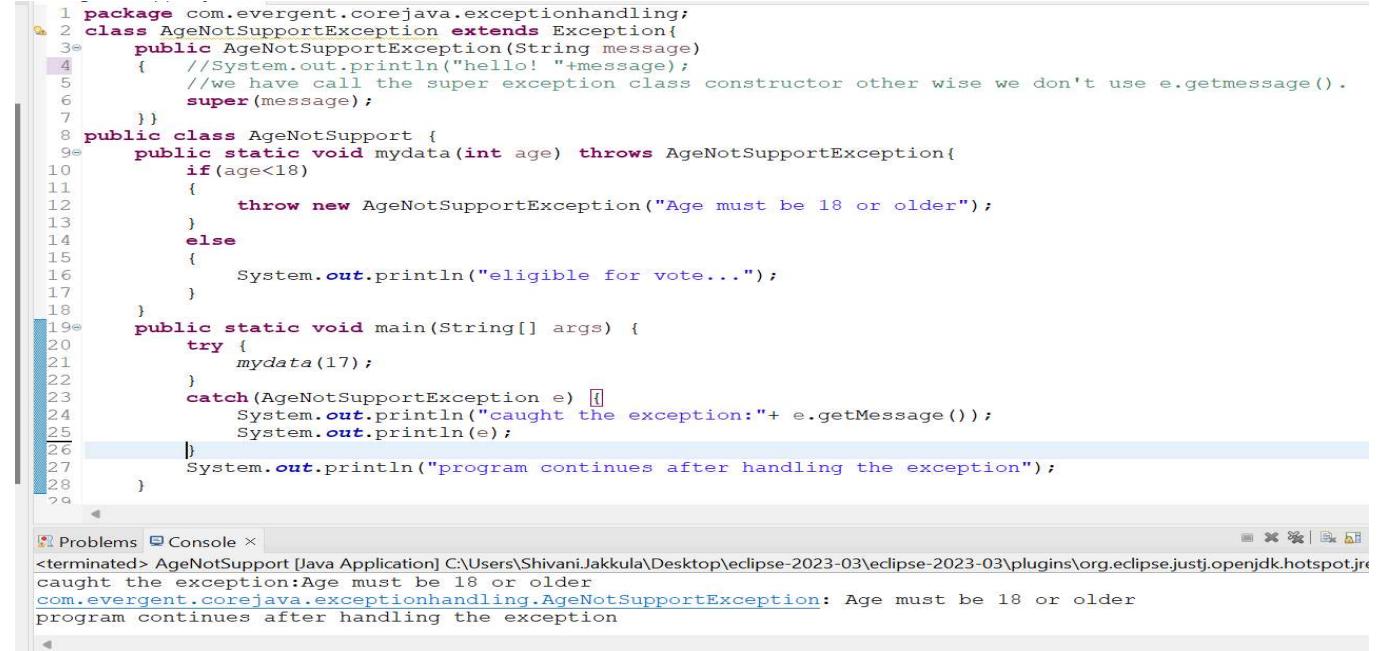
```
package com.evergent.corejava.exceptionhandling;

class ProductNotFoundException extends Exception{
    public ProductNotFoundException(String message)
    {
        System.out.println("hello"+message);
    }
}
public class ProductImpl {
    int pno=105;
    public void mydata() throws ProductNotFoundException{
        if(pno>100)
        {
            throw new ProductNotFoundException("there is no products");
        }
        else
        {
            System.out.println("product are there..");
        }
    }
    public static void main(String[] args) {
        try {
            ProductImpl pl=new ProductImpl();
            pl.mydata();
        }
        catch(ProductNotFoundException e) {
            System.out.print(e);
        }
    }
}
```

```
<terminated> ProductImpl (1) [Java Application] C:\Users\Shivani.Jakkula\Desktop\java-2023
hellothere is no products
com.evergent.corejava.exceptionhandling.ProductNotFoundException: there is no products
```

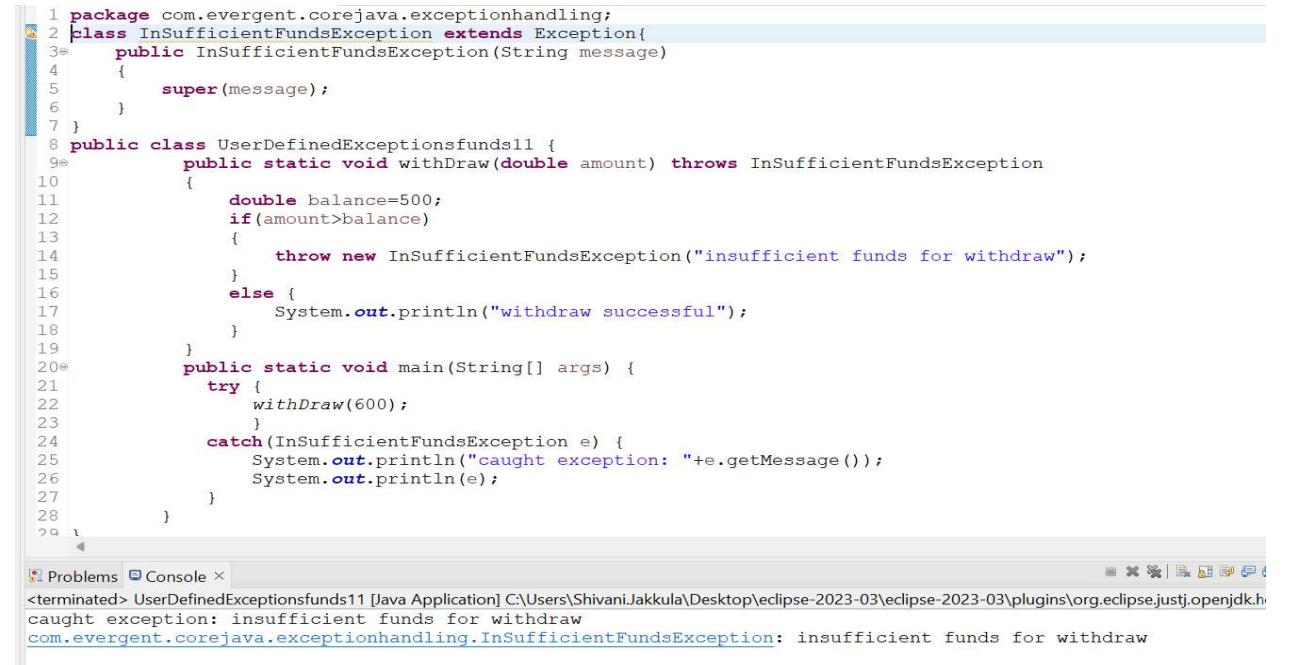
21/08/2024- Day 12

Program 10:



```
1 package com.evergent.corejava.exceptionhandling;
2 class AgeNotSupportException extends Exception{
3     public AgeNotSupportException(String message)
4     {
5         //System.out.println("hello! "+message);
6         //we have call the super exception class constructor other wise we don't use e.getMessage().
7         super(message);
8     }
9     public class AgeNotSupport {
10         public static void mydata(int age) throws AgeNotSupportException{
11             if(age<18)
12             {
13                 throw new AgeNotSupportException("Age must be 18 or older");
14             }
15             {
16                 System.out.println("eligible for vote...");
17             }
18         }
19         public static void main(String[] args) {
20             try {
21                 mydata(17);
22             }
23             catch(AgeNotSupportException e) {
24                 System.out.println("caught the exception:"+ e.getMessage());
25                 System.out.println(e);
26             }
27             System.out.println("program continues after handling the exception");
28         }
29     }
}
Problems Console <
<terminated> AgeNotSupport [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jre
caught the exception:Age must be 18 or older
com.evergent.corejava.exceptionhandling.AgeNotSupportException: Age must be 18 or older
program continues after handling the exception
```

Program 11:



```
1 package com.evergent.corejava.exceptionhandling;
2 class InSufficientFundsException extends Exception{
3     public InSufficientFundsException(String message)
4     {
5         super(message);
6     }
7     public class UserDefinedExceptionsfunds11 {
8         public static void withDraw(double amount) throws InSufficientFundsException
9         {
10             double balance=500;
11             if(amount>balance)
12             {
13                 throw new InSufficientFundsException("insufficient funds for withdraw");
14             }
15             else {
16                 System.out.println("withdraw successful");
17             }
18         }
19         public static void main(String[] args) {
20             try {
21                 withDraw(600);
22             }
23             catch(InSufficientFundsException e) {
24                 System.out.println("caught exception: "+e.getMessage());
25                 System.out.println(e);
26             }
27         }
28     }
}
Problems Console <
<terminated> UserDefinedExceptionsfunds11 [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.h
caught exception: insufficient funds for withdraw
com.evergent.corejava.exceptionhandling.InSufficientFundsException: insufficient funds for withdraw
```

Program 12:

```
1 package com.evergent.corejava.exceptionhandling;
2 class InvalidScoreException extends RuntimeException{
3     public InvalidScoreException(String message)
4     {
5         super(message);
6     }
7     public class UserDefinedUncheckedexceptiondemo12 {
8         public void validateScore(int score)
9         {
10             if(score<0 || score>100)
11             {
12                 throw new InvalidScoreException("score must be between 0 and 100..");
13             }
14             else {
15                 System.out.println("valid score...");}
16         }
17     }
18     public static void main(String[] args) {
19         try {
20             UserDefinedUncheckedexceptiondemo12 udu=new UserDefinedUncheckedexceptiondemo12 ();
21             udu.validateScore(110);
22         }
23         catch(InvalidScoreException e)
24         {
25             System.out.println("caught exception: "+e.getMessage());
26             System.out.println(e);
27         }
28         System.out.println("program continues after handling...");}
29 }
```

Console output:

```
Problems Console ×
<terminated> UserDefinedUncheckedexceptiondemo12 [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\com.evergent.corejava.exceptionhandling.InvalidScoreException: score must be between 0 and 100..
caught exception: score must be between 0 and 100..
com.evergent.corejava.exceptionhandling.InvalidScoreException: score must be between 0 and 100..
program continues after handling...
```

Program 13:

```
1 package com.evergent.corejava.exceptionhandling;
2
3 public class ArrayIndexOutOfBoundsException13 {
4     public static void main(String[] args) {
5         int[] arr= {1,2,3,4,5};
6         try {
7             System.out.println(arr[10]);
8         }
9         catch(ArrayIndexOutOfBoundsException e)
10        {
11            System.out.println(e.getMessage());
12        }
13    }
14
15 }
16
```

Console output:

```
Problems Console ×
<terminated> ArrayIndexOutOfBoundsException13 [Java Application] C:\Users\Shivani.Jakkula\Index 10 out of bounds for length 5
```

Program 14:

```
1 package com.evergent.corejava.exceptionhandling;
2
3 public class CommandLineArguments {
4     public static void main(String[] args) {
5         try {
6             System.out.println(args[4]);
7         }
8         catch (ArrayIndexOutOfBoundsException e)
9         {
10             System.out.println("no such index argument");
11         }
12     }
13 }
```

Problems Console ×
<terminated> CommandLineArguments [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\workspace\Java\src\com\evergent\corejava\exceptionhandling\CommandLineArguments.java:12: error: no such index argument

Program 15:

```
1 package com.evergent.corejava.exceptionhandling;
2 import java.io.*;
3 import java.util.Scanner;
4 public class CompileTimeFileDemo15 {
5     public static void main(String[] args) {
6         try {
7             File file=new File("C:/mydata/myinfo.txt");
8             Scanner sc=new Scanner(file);
9             while(sc.hasNextLine()) {
10                 System.out.println(sc.nextLine());
11             }
12         }
13         catch (Exception e)
14         {
15             System.out.println(e);
16         }
17     }
18 }
```

Problems Console ×
<terminated> CompileTimeFileDemo15 [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\workspace\Java\src\com\evergent\corejava\exceptionhandling\CompileTimeFileDemo15.java:15: error: java.io.FileNotFoundException: C:\mydata\myinfo.txt (The system cannot find the path specified)

JAVABEANS:

- Javabeans is a mechanism.
- Javabeans is lightweight.
- All attributes are private and get/set methods are public implements java.io.Serializable interface.
- We can achieve tightly encapsulation through javabeans.

Program1:

The screenshot shows the Eclipse IDE interface with two code editors and a console window. The left code editor contains `EmployeeImpl.java` and the right one contains `Employee.java`. The console window shows the output of the `EmployeeImpl` application.

```
EmployeeImpl.java
1 package com.evergent.corejava.javabeans;
2
3 public class EmployeeImpl {
4     public static void main(String[] args) {
5         //Initializing and retriving through getter and
6         Employee emp=new Employee();
7         emp.setEno(10);
8         emp.setEname("shivani");
9         emp.setSal(50000);
10        System.out.println(emp.getEno());
11        System.out.println(emp.getEname());
12        System.out.println(emp.getSal());
13        emp.setEno(100);
14        System.out.println(emp.getEno());
15    }
16 }
```

```
Employee.java
1 package com.evergent.corejava.javabeans;
2 import java.io.Serializable;
3 public class Employee implements Serializable{
4     private int eno;
5     public int getEno() {
6         return eno;
7     }
8     public void setEno(int eno) {
9         this.eno = eno;
10    }
11    public String getEname() {
12        return ename;
13    }
14    public void setEname(String ename) {
15        this.ename = ename;
16    }
17    public double getSal() {
18        return sal;
19    }
20    public void setSal(double sal) {
21        this.sal = sal;
22    }
23    private String ename;
24    private double sal;
25 }
```

```
Console
<terminated> EmployeeImpl [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-07-14
10
shivani
50000.0
100
```

Program2:

The screenshot shows the Eclipse IDE interface with two code editors and a console window. The left code editor contains `ProductImpl.java` and the right one contains `Product.java`. The console window shows the output of the `ProductImpl` application.

```
ProductImpl.java
1 package com.evergent.corejava.javabeans;
2
3 public class ProductImpl {
4     public static void main(String[] args) {
5         Product p=new Product(1,"Laptop",61000);
6         System.out.println(p.getPno());
7         System.out.println(p.getPname());
8         System.out.println(p.getPrice());
9     }
10 }
11
12 
```

```
Product.java
1 package com.evergent.corejava.javabeans;
2 import java.io.Serializable;
3 public class Product implements Serializable{
4     private int pno;
5     private String pname;
6     private double price;
7
8     public Product(int pno, String pname, double price)
9         this.pno = pno;
10        this.pname = pname;
11        this.price = price;
12    }
13    public int getPno() {
14        return pno;
15    }
16    public String getPname() {
17        return pname;
18    }
19    public double getPrice() {
20        return price;
21    }
22 }
```

```
Console
<terminated> ProductImpl [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-07-14
1
Laptop
61000.0
```

Program 3:

The screenshot shows the Eclipse IDE interface with three code editors and a console window. The left code editor contains `StudentImpl.java`, the middle one contains `Product.java`, and the right one contains `Student.java`. The console window shows the output of the `StudentImpl` application.

```
StudentImpl.java
1 package com.evergent.corejava.javabeans;
2 import java.io.Serializable;
3 public class StudentImpl implements Serializable {
4     public static void main(String[] args) {
5         Student s=new Student();
6         s.setSno(1);
7         s.setName("Shivani");
8         s.setAdd("Hyderabad");
9         System.out.println(s);
10    }
11 }
12
13 }
```

```
Product.java
1 package com.evergent.corejava.javabeans;
2
3 public class Product {
4     private int pno;
5     private String pname;
6     private double price;
7
8     public Product(int pno, String pname, double price)
9         this.pno = pno;
10        this.pname = pname;
11        this.price = price;
12    }
13    public int getPno() {
14        return pno;
15    }
16    public String getPname() {
17        return pname;
18    }
19    public double getPrice() {
20        return price;
21    }
22 }
```

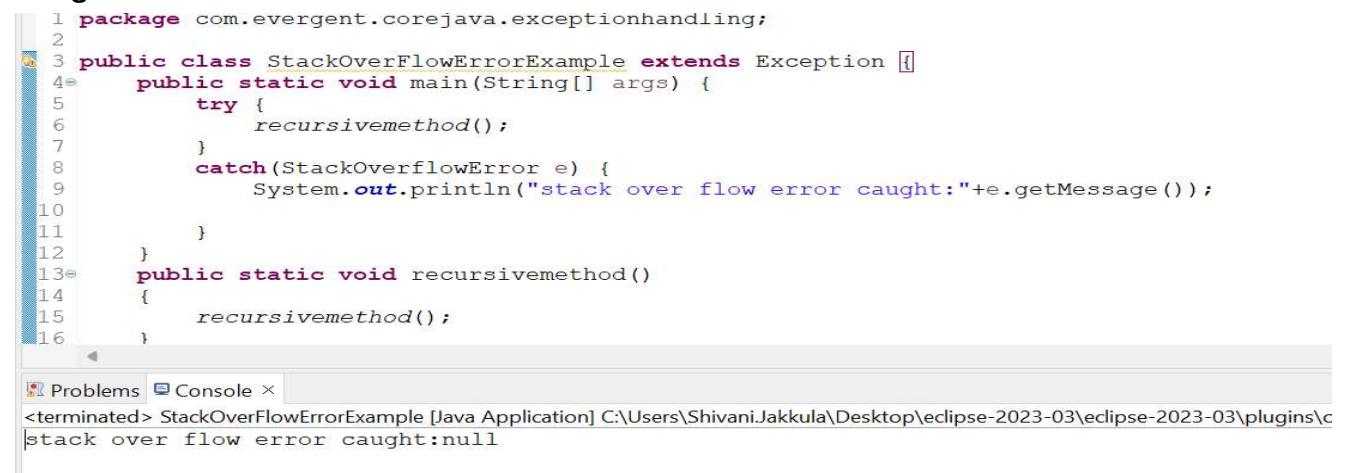
```
Student.java
1 package com.evergent.corejava.javabeans;
2
3 public class Student {
4     private int sno;
5     private String name;
6     private String add;
7     public void setSno(int sno) {
8         this.sno = sno;
9     }
10    public void setName(String name) {
11        this.name = name;
12    }
13    public void setAdd(String add) {
14        this.add = add;
15    }
16    @Override
17    public String toString() {
18        return "num of student: "+sno+"\n name of student: "+name+"\n address of student: "+add;
19    }
20 }
21
22 }
```

```
Console
<terminated> StudentImpl [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-07-14
num of student: 1
name of student: Shivani
Address of student: Hyderabad
```

22/08/2024- Day 13

Errors:

Program1:

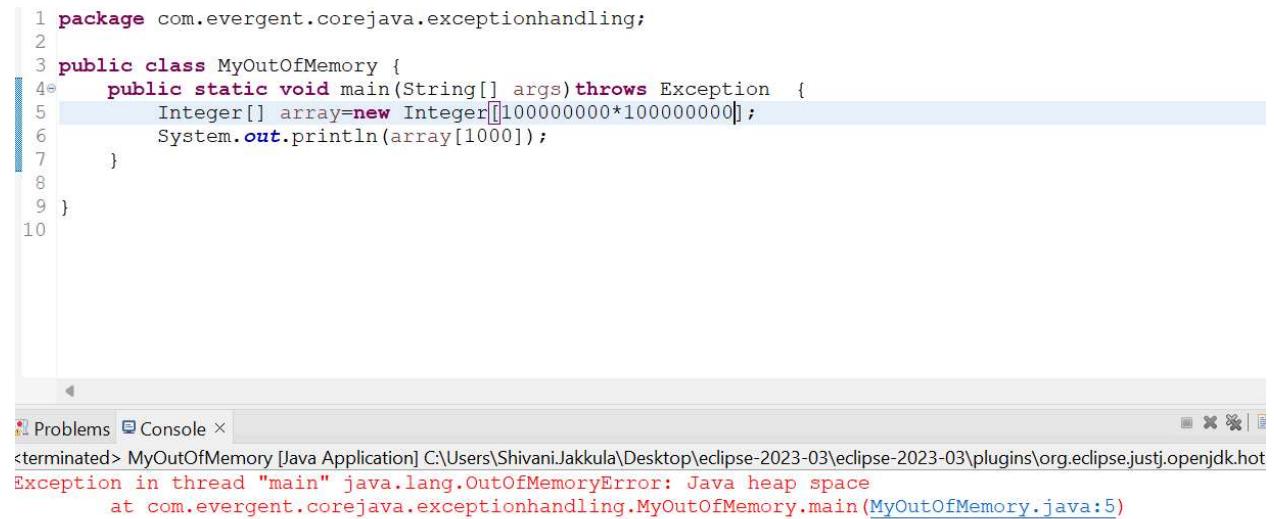


```
1 package com.evergent.corejava.exceptionhandling;
2
3 public class StackOverflowErrorExample extends Exception {
4     public static void main(String[] args) {
5         try {
6             recursivemethod();
7         } catch(StackOverflowError e) {
8             System.out.println("stack over flow error caught:"+e.getMessage());
9         }
10    }
11 }
12 public static void recursivemethod() {
13 {
14     recursivemethod();
15 }
16 }
```

Problems Console ×

<terminated> StackOverflowErrorExample [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\com.evergent.corejava.exceptionhandling\src\main\java\com\evergent\corejava\exceptionhandling\StackOverflowErrorExample.java:16: stack over flow error caught:null

Program2:



```
1 package com.evergent.corejava.exceptionhandling;
2
3 public class MyOutOfMemory {
4     public static void main(String[] args) throws Exception {
5         Integer[] array=new Integer[100000000*10000000];
6         System.out.println(array[1000]);
7     }
8
9 }
10
```

Problems Console ×

<terminated> MyOutOfMemory [Java Application] C:\Users\Shivani.Jakkula\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jdk17\src\main\java\com\evergent\corejava\exceptionhandling\MyOutOfMemory.java:5: Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
at com.evergent.corejava.exceptionhandling.MyOutOfMemory.main (MyOutOfMemory.java:5)