CORE JAVA DEVELOPMENT

Date:05/08/2024
Day 1
1.Language and Applications

2.Java Features
- Why java is Platform Independent?
- OOPS
- Exceptional Handling
- Multi Threading
- Security
- Open Source
- Networking
- Memory Management
- Web Application

3.JDK,JRE,JVM

4.Basic Java Programming
- Variables
- Methods
- Conditional Statements
- Loops

5.Packages

Date:06-08-2024
Day 2
1.Loops

2.Single Dimensional Arrays

3.Two Dimensional Arrays

4.Logical Programs
- Fibnocci
- Factorial

5.Switch Case
6. Enum
7. Scanner class
8. Object class and methods
9. Packages
10. Event Management System(EMS) Application overview

Date:07-08-2024
Day 3

1.OOPS

- Encapsulation
    - ➔ Calculation program
    - ➔ Person info program
    - ➔ Methods flow program
- Polymorphism
    - ➔ Method Overloading
    - ➔ Method Overriding
- Inheritance
    - ➔ Multiple
    - ➔ Multilevel
- Abstraction

DAY 6:

Date:12-08-2024

Strings are there in java.lang package

1.String Class

- String is Final Class
- String class has Methods
- String Methods are Non Synchronized
- String is Immutable Class
- Methods
    - equals()
    - toUpperCase()
    - toLowerCase()
    - trim()
    - length()

2.String Buffer

- String Buffer is Final Class
- String Buffer is Mutable
- String Buffer having Methods
- All string methods are Synchronized
- String buffer is not recommended for development but still in API because it's outdated.
- String Buffer is Thread Safe(Synchronization)
- Methods
    - append()
    - insert()
    - replace()
    - delete()
    - reverse()
    - capacity()
    - length()

3.String Builder (1.5 version) Tiger Version

- String Builder is Final Class
- String Builder is Mutable Class
- String Builder methods are Non Synchronized
- String Builder having methods
- Methods
    - append()
    - insert()
    - delete()
    - reverse()
    - replace()

4.**Strings are Immutable** because when we are trying to create is a String Object(new keyword) it is constant.If we are trying to modify existing String it will create another memory location of the String ,then the Existing object is Eligible for Garbage Collection.

5.In java String is a Sequence of characters used to represent text

6.These are part of java.lang package

7.Immutable strings:Once the object is created it can't be changed.Any modifications string create a new string object.

8.Strings are stored String pool memory

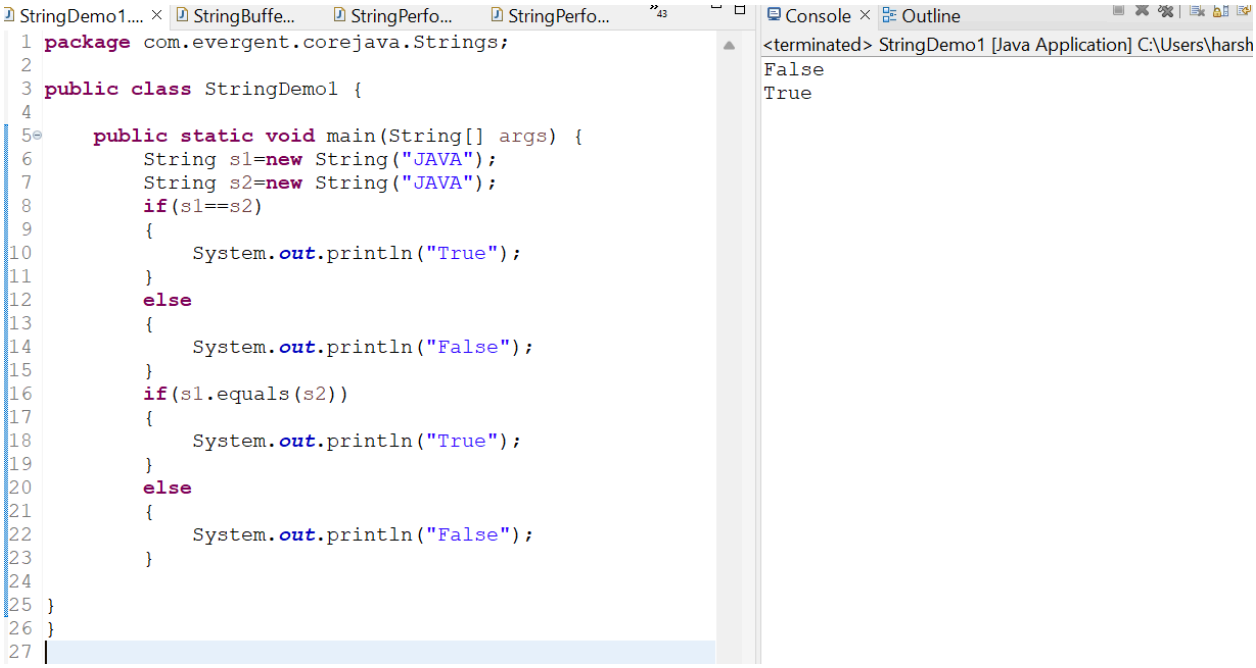
9.If two strings are created without new and with same content then they will reference the same object in the string pool.

10.Ways of creating String

- Using **string literals**
  String str="Hello world";
- Using **new** keyword
  String str=new String("Hello World");

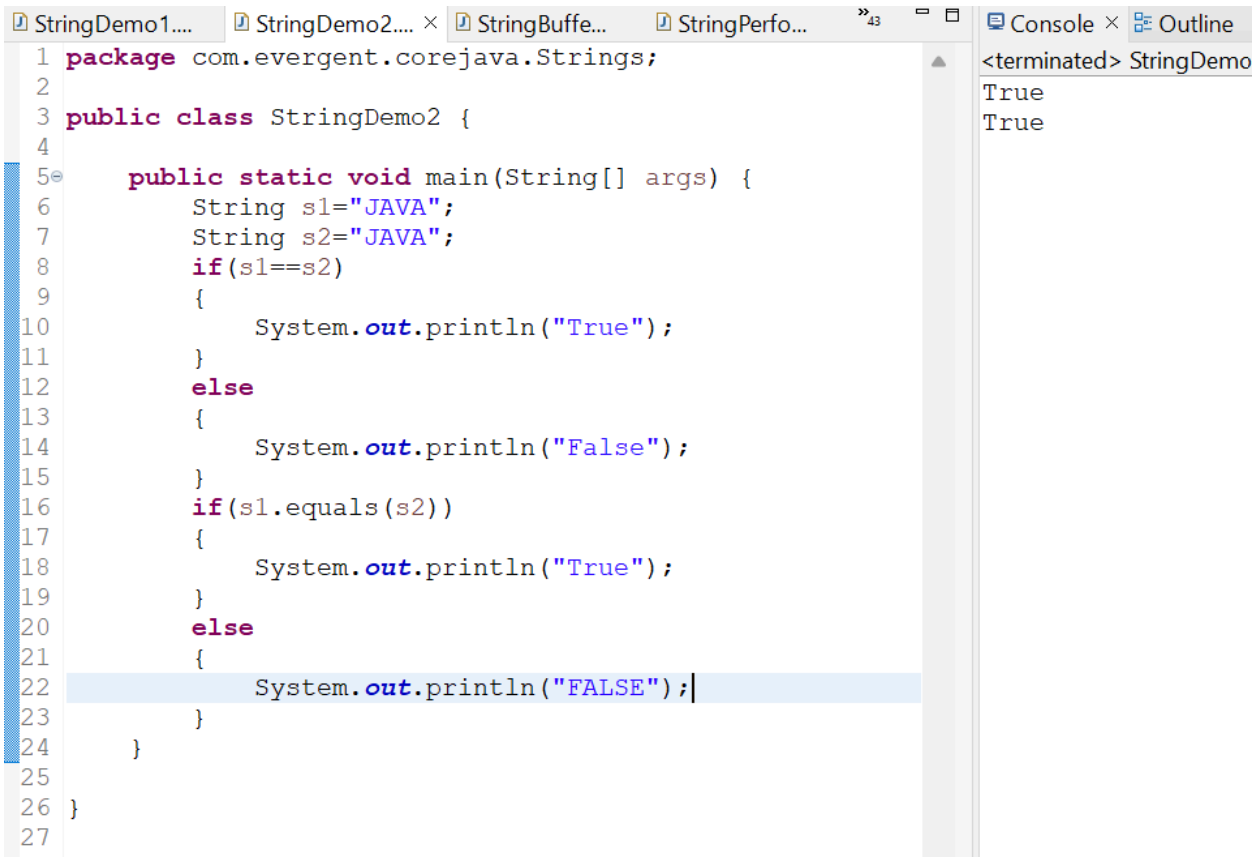
PROGRAMS:

Program1:




```
1  package com.evergent.corejava.Strings;
2
3  public class StringDemo1 {
4
5      public static void main(String[] args) {
6          String s1=new String("JAVA");
7          String s2=new String("JAVA");
8          if(s1==s2)
9          {
10             System.out.println("True");
11         }
12         else
13         {
14             System.out.println("False");
15         }
16         if(s1.equals(s2))
17         {
18             System.out.println("True");
19         }
20         else
21         {
22             System.out.println("False");
23         }
24
25 }
26 }
27
```

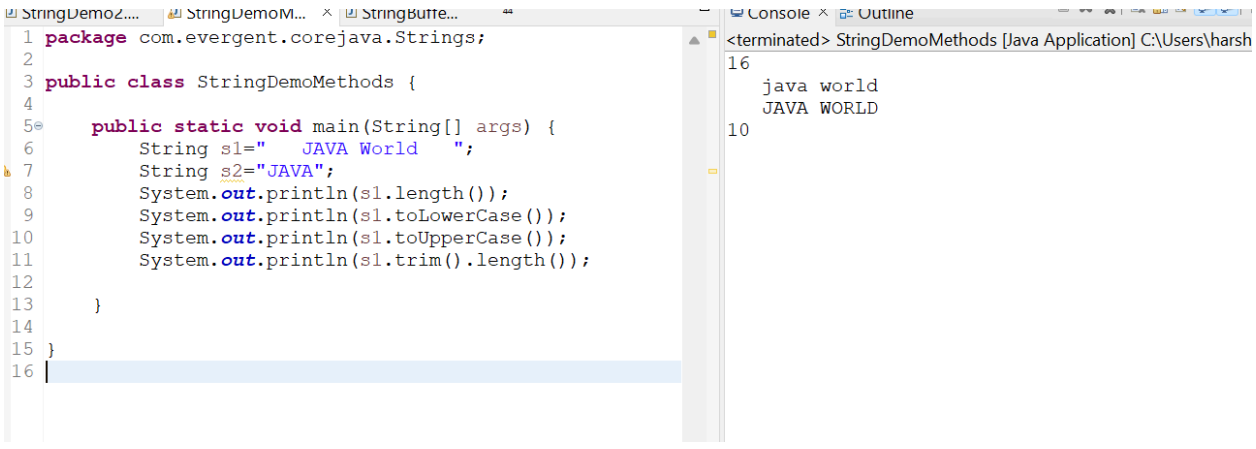
Program2:



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

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");
        }
    }

}
```

```
<terminated> StringDemo
True
True
```

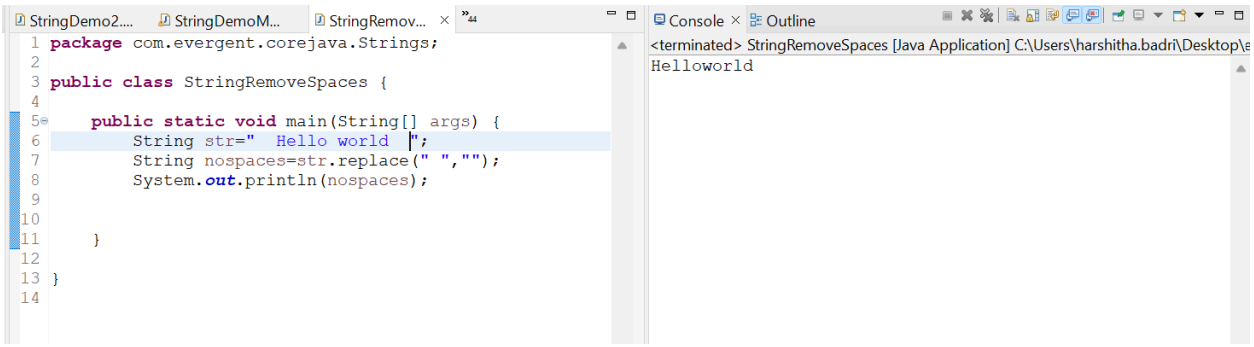
Program 3:



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

public class StringDemoMethods {

    public static void main(String[] args) {
        String s1="   JAVA World   ";
        String s2="JAVA";
        System.out.println(s1.length());
        System.out.println(s1.toLowerCase());
        System.out.println(s1.toUpperCase());
        System.out.println(s1.trim().length());

    }

}
```

```
<terminated> StringDemoMethods [Java Application] C:\Users\harsh
16
   java world
   JAVA WORLD
10
```

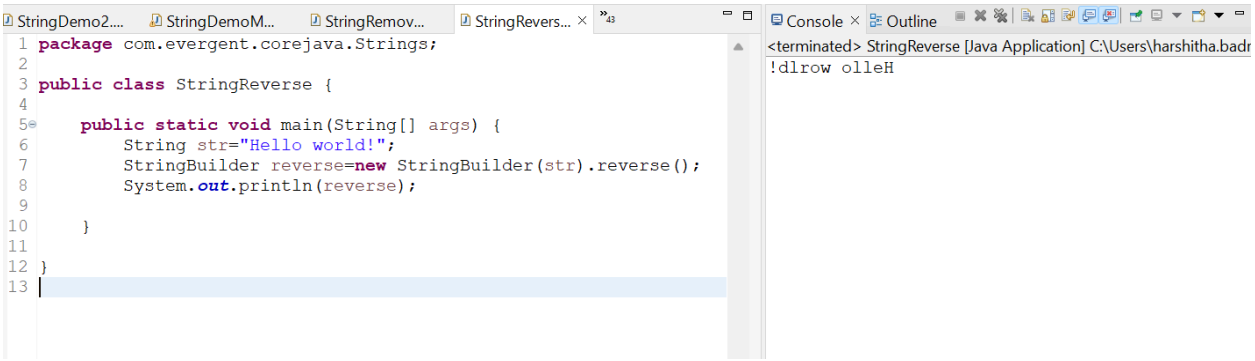
Program 4:



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

public class StringRemoveSpaces {

    public static void main(String[] args) {
        String str="  Hello world  ";
        String nospaces=str.replace(" ","");
        System.out.println(nospaces);


    }

}
```

```
<terminated> StringRemoveSpaces [Java Application] C:\Users\harshitha.badri\Desktop\e
Helloworld
```

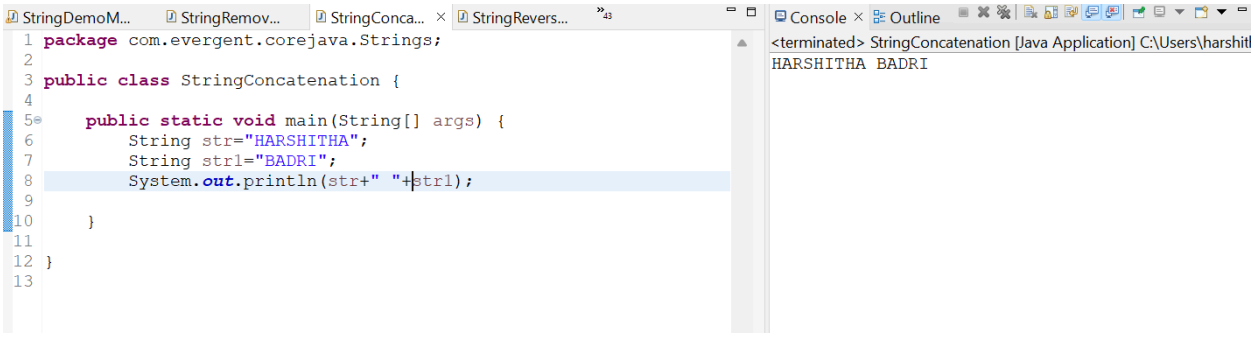
Program 5:



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

public class StringReverse {

    public static void main(String[] args) {
        String str="Hello world!";
        StringBuilder reverse=new StringBuilder(str).reverse();
        System.out.println(reverse);

    }

}
```

```
<terminated> StringReverse [Java Application] C:\Users\harshitha.badr
!dlrow olleH
```

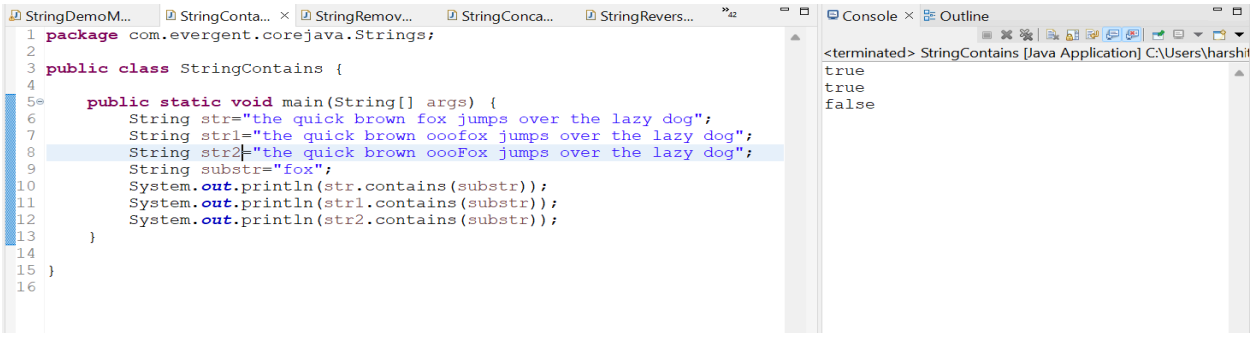
Program 6:



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

public class StringConcatenation {

    public static void main(String[] args) {
        String str="HARSHITHA";
        String str1="BADRI";
        System.out.println(str+" "+str1);

    }

}
```

```
<terminated> StringConcatenation [Java Application] C:\Users\harshitl
HARSHITHA BADRI
```

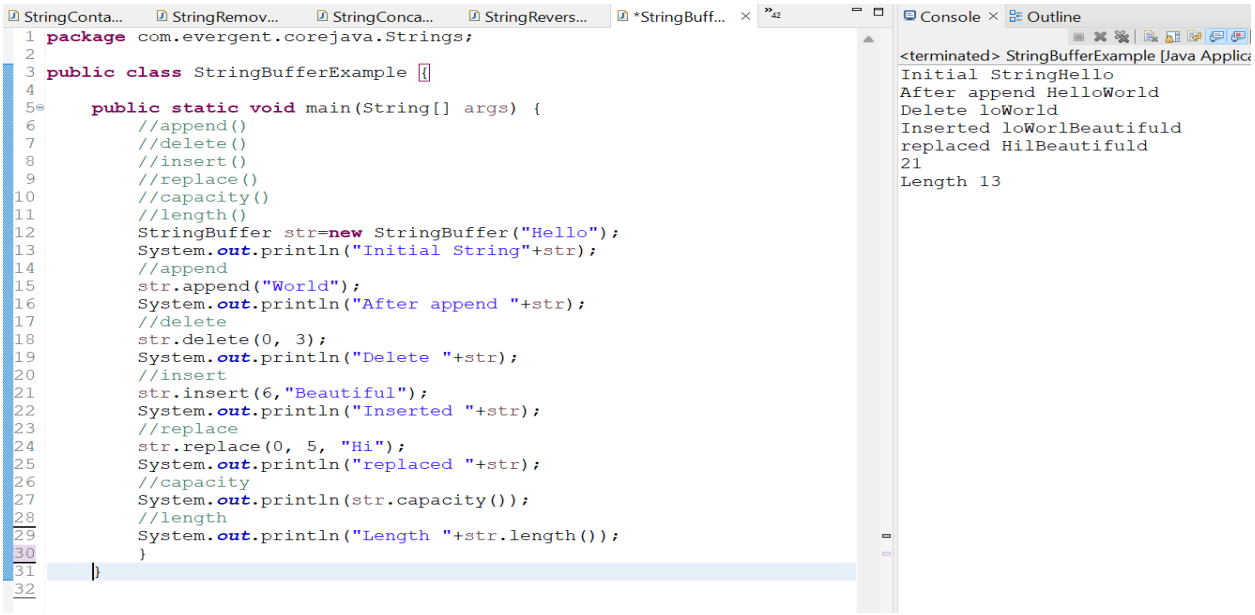
Program 7:



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

public class StringContains {

    public static void main(String[] args) {
        String str="the quick brown fox jumps over the lazy dog";
        String str1="the quick brown ooofox jumps over the lazy dog";
        String str2="the quick brown oooFox jumps over the lazy dog";
        String substr="fox";
        System.out.println(str.contains(substr));
        System.out.println(str1.contains(substr));
        System.out.println(str2.contains(substr));
    }

}
```

```
<terminated> StringContains [Java Application] C:\Users\harshi
true
true
false
```

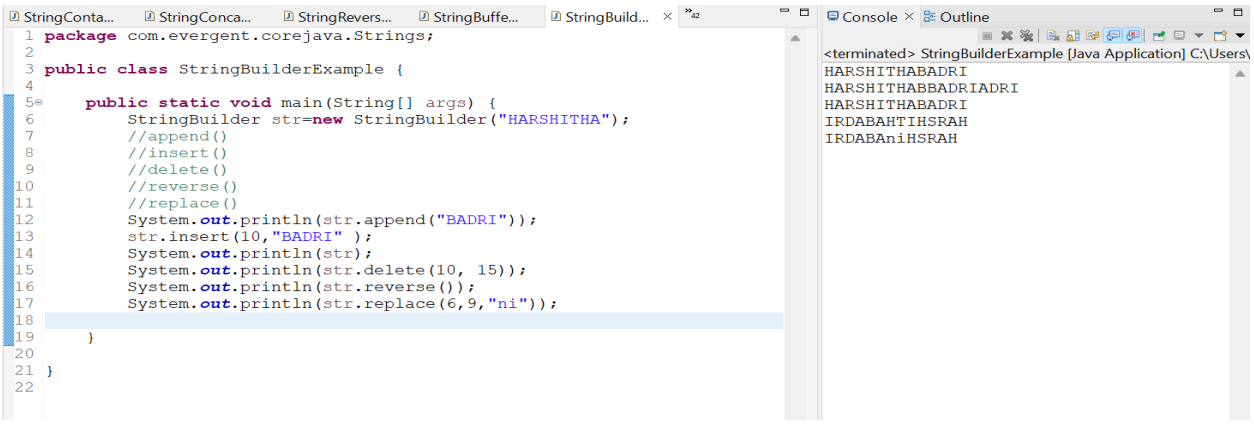
Program 8:



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

public class StringBufferExample {

    public static void main(String[] args) {
        //append()
        //delete()
        //insert()
        //replace()
        //capacity()
        //length()
        StringBuffer str=new StringBuffer("Hello");
        System.out.println("Initial String"+str);
        //append
        str.append("World");
        System.out.println("After append "+str);
        //delete
        str.delete(0, 3);
        System.out.println("Delete "+str);
        //insert
        str.insert(6,"Beautiful");
        System.out.println("Inserted "+str);
        //replace
        str.replace(0, 5, "Hi");
        System.out.println("replaced "+str);
        //capacity
        System.out.println(str.capacity());
        //length
        System.out.println("Length "+str.length());
        }
    }
```

```
<terminated> StringBufferExample [Java Applica
Initial StringHello
After append HelloWorld
Delete loWorld
Inserted loWorlBeautifuld
replaced HilBeautifuld
21
Length 13
```

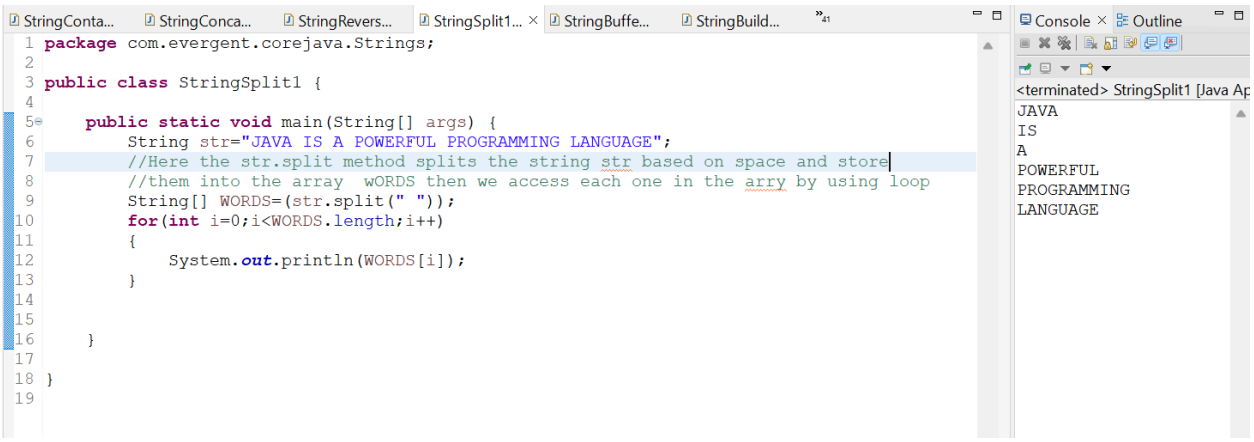
Program 9:



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

public class StringBuilderExample {

    public static void main(String[] args) {
        StringBuilder str=new StringBuilder("HARSHITHA");
        //append()
        //insert()
        //delete()
        //reverse()
        //replace()
        System.out.println(str.append("BADRI"));
        str.insert(10,"BADRI" );
        System.out.println(str);
        System.out.println(str.delete(10, 15));
        System.out.println(str.reverse());
        System.out.println(str.replace(6,9,"ni"));

    }

}
```

```
<terminated> StringBuilderExample [Java Application] C:\Users\
HARSHITHABADRI
HARSHITHABBADRIADRI
HARSHITHABADRI
IRDABAHTIHSRAH
IRDABAniHSRAH
```

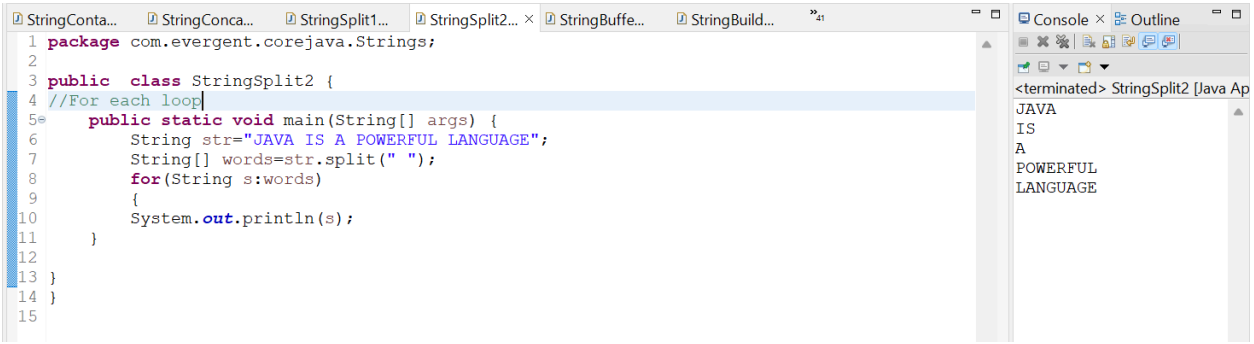
Program 10:



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

public class StringSplit1 {

    public static void main(String[] args) {
        String str="JAVA IS A POWERFUL PROGRAMMING LANGUAGE";
        //Here the str.split method splits the string str based on space and store
        //them into the array  wORDS then we access each one in the arry by using loop
        String[] WORDS=(str.split(" "));
        for(int i=0;i<WORDS.length;i++)
        {
            System.out.println(WORDS[i]);
        }


    }

}
```

```
<terminated> StringSplit1 [Java Ap
JAVA
IS
A
POWERFUL
PROGRAMMING
LANGUAGE
```

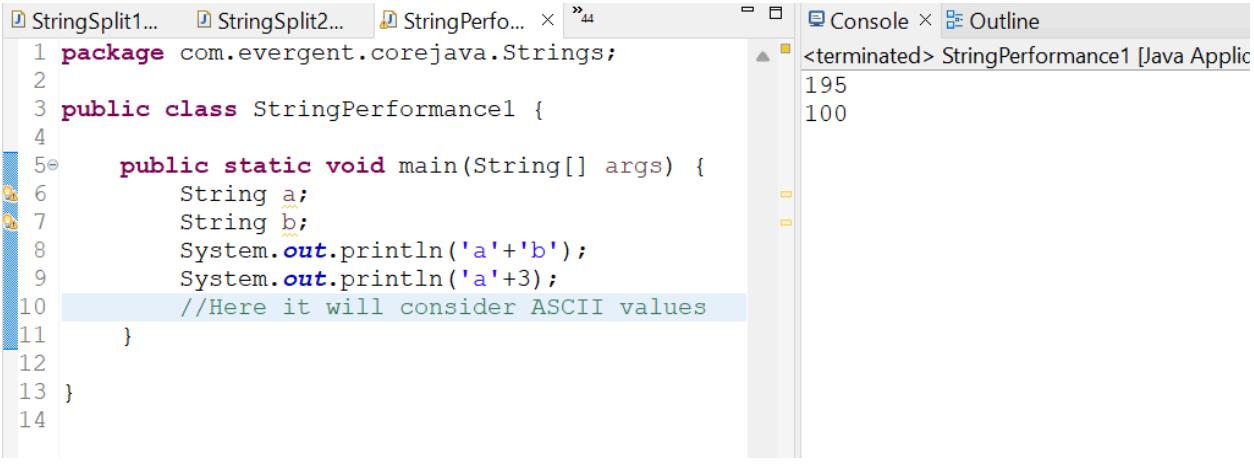
Program 11:



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

public  class StringSplit2 {
//For each loop
    public static void main(String[] args) {
        String str="JAVA IS A POWERFUL LANGUAGE";
        String[] words=str.split(" ");
        for(String s:words)
        {
        System.out.println(s);
    }

}
}
```

```
<terminated> StringSplit2 [Java Ap
JAVA
IS
A
POWERFUL
LANGUAGE
```

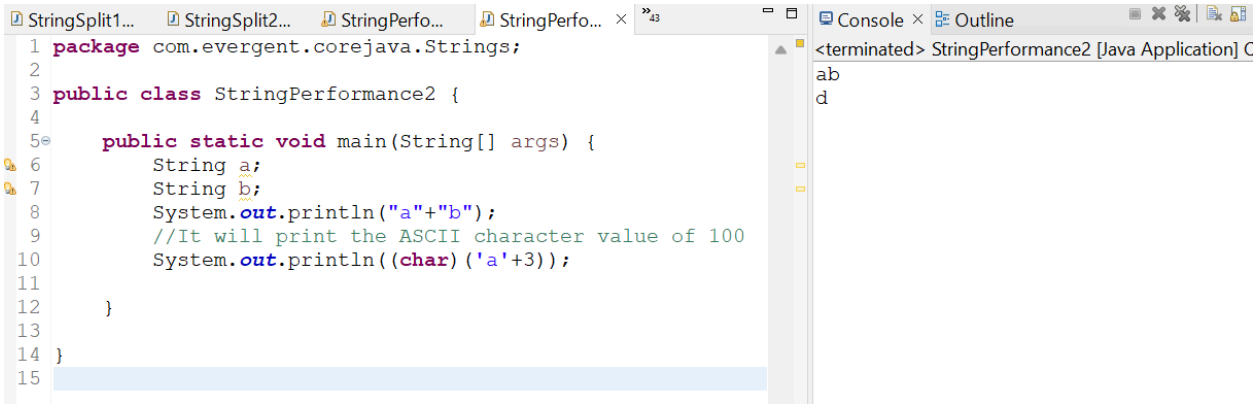
Program 12:



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

public class StringPerformance1 {

    public static void main(String[] args) {
        String a;
        String b;
        System.out.println('a'+'b');
        System.out.println('a'+3);
        //Here it will consider ASCII values
    }

}
```

```
<terminated> StringPerformance1 [Java Applic
195
100
```

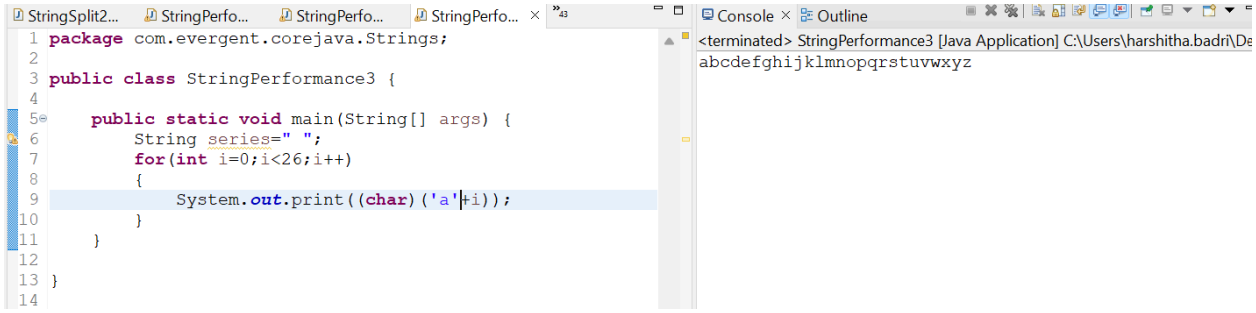
Program 13:



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

public class StringPerformance2 {

    public static void main(String[] args) {
        String a;
        String b;
        System.out.println("a"+"b");
        //It will print the ASCII character value of 100
        System.out.println((char)('a'+3));

    }

}
```

```
ab
d
```

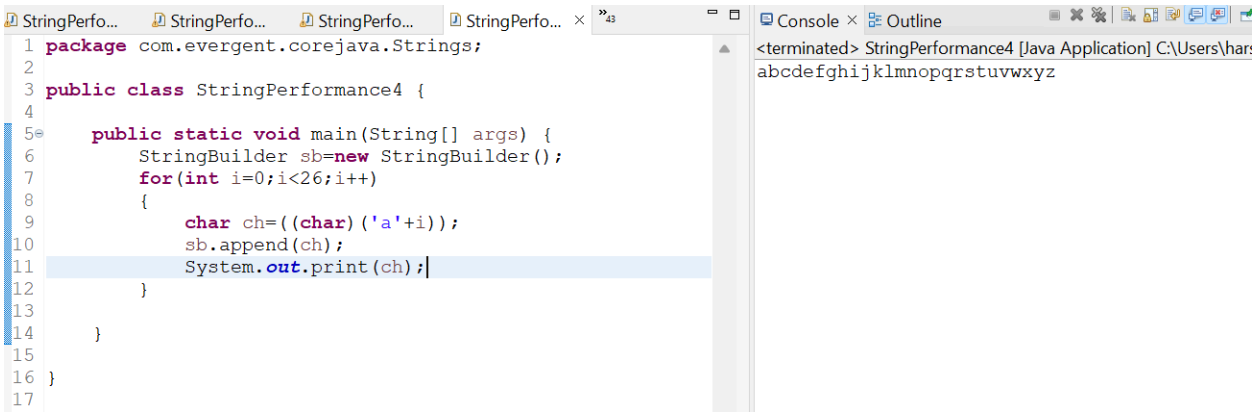
Program 14:



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

public class StringPerformance3 {

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

}
```

```
abcdefghijklmnopqrstuvwxyz
```

Program 15:



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

public class StringPerformance4 {

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

    }

}
```

```
abcdefghijklmnopqrstuvwxyz
```

Program 16:

```java
package com.evergent.corejava.Strings;
import java.util.Arrays;
public class StringPerformance5 {

    public static void main(String[] args) {
        String str="Java is a Platform independent ";

        System.out.println(Arrays.toString(str.toCharArray()));
    }

}
```

```
<terminated> StringPerformance5 [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.h
[J, a, v, a,  , i, s,  , a,  , P, l, a, t, f, o, r, m,  , i, n, d, e, p, e, n, d, e, n, t,  ]
```

Day:7
Date:13-08-2024

STRING IMMUTABLE PROGRAMS:

1. Strings are Immutable.We can make class as immutable by declaring it as Final class and declare variables as private and final.
2. Class is declared as final so that it cannot be subclassed.
3. While calling constructor we can initialize final variable.

PROGRAM 1:

```java
package com.evergent.corejava.StringsImmutable;

public final class PersonImmutable {

    private final String name;
    private final int age;
    public PersonImmutable(String name,int age)
    {
        this.name=name;
        this.age=age;
    }
    public String myName()
    {
        return name;
    }
    public int myAge()
    {
        return age;
    }


}
```

MAIN PROGRAM OF ABOVE PROGRAM:

```java
package com.evergent.corejava.StringsImmutable;

public class MainPerson {

    public static void main(String[] args) {
        PersonImmutable mp=new PersonImmutable("HARSHITHA",21);
        System.out.println(mp.myName());
        System.out.println(mp.myAge());

    }

}
```

```
<terminated> MainPerson
HARSHITHA
21
```

PROGRAM 2:
Tostring method

```java
package com.evergent.corejava.StringsImmutable;

public final class ImmutableString {
    private final String value;
    public ImmutableString(String value)
    {
        this.value=value;
    }
    public String myValue()
    {
        return value;
    }
    public String toString()
    {
        return value;
    }

    public static void main(String[] args) {
        ImmutableString iss=new ImmutableString("HARSHITHA");
        System.out.println(iss.myValue());
        System.out.println(iss.toString());

    }

}
```

```
<terminated> ImmutableString [Ja
HARSHITHA
HARSHITHA
```

OBJECT CLASS METHODS:
HASHCODE METHOD:

BookImpl4.java | PersonImmut... | MainPerson.java | ImmutableSt... | ObjectClass...

```java
package com.evergent.corejava.ObjectClassMethods;

public class ObjectClassMethods {
    String name;
    int age;
    public ObjectClassMethods (String name,int age)
    {
        this.name=name;
        this.age=age;
    }
    public String toString()
    {
        return "Name:"+name+"  \n Age:"+age;
    }

    public static void main(String[] args) {
        ObjectClassMethods oc=new ObjectClassMethods("harshitha",21);
        System.out.println(oc);
        System.out.println(oc.hashCode());
    }
}
```

Outline | Console

```
<terminated> ObjectClassMethods [Java App
Name:harshitha
 Age:21
2083562754
```

INTERFACES:

1. Interface is a **Keyword**
2. We can declare method signatures only but not implementation
3. By default all interfaces methods are **Abstract**
4. If any class implements interface,that class should **override all interface methods** otherwise the class is showing **compile time error.**
5. We cannot create object to interface but we can create reference to interface -*Class implements interface program 2*
6. The **variables** inside the interfaces are **public,static,final**
7. Java support **multiple inheritance** through **Intefaces**
8. One **class** can **implements** more than one **interface**
9. One **interface** can **extends** other **interfaces**

INTERFACE PROGRAMS :

Interface Program 1:

```
1  package com.evergent.corejava.Interface;
2
3  public interface Book {
4      //By default the variables inside the interface are public static and final
5      public void bookTitle();
6      public void bookAuthor();
7      public void bookPrice();
8
9  }
10
```

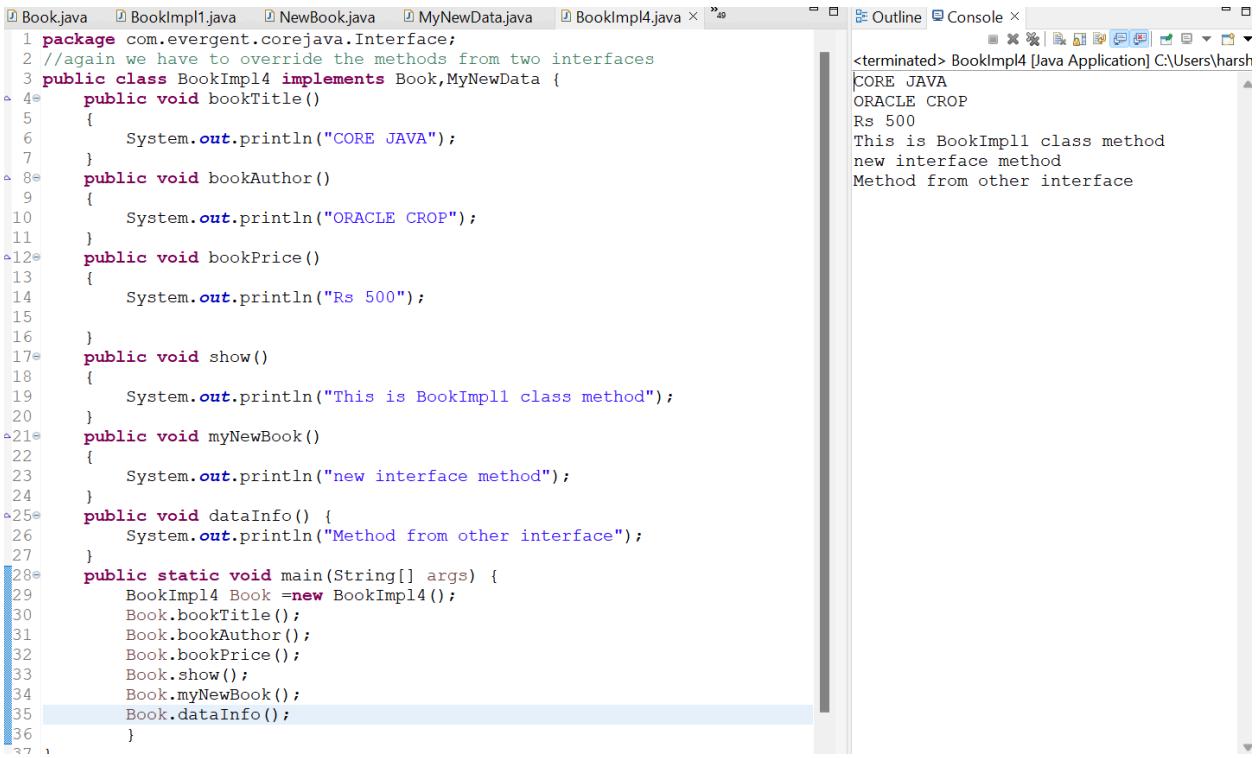
Interface Program 2:

```
1  package com.evergent.corejava.Interface;
2
3  public interface NewBook {
4      public void myNewBook();
5      public void bookPrice();
6
7  }
8
```

Interface Program 3:

```java
package com.evergent.corejava.Interface;
//Two interfaces extends each other
public interface MyNewData extends NewBook{
    public void dataInfo();

}

```

Class Implements Interface Program 1:



```java
package com.evergent.corejava.Interface;
//again we have to override the methods from two interfaces
public class BookImpl4 implements Book,MyNewData {
    public void bookTitle()
    {
        System.out.println("CORE JAVA");
    }
    public void bookAuthor()
    {
        System.out.println("ORACLE CROP");
    }
    public void bookPrice()
    {
        System.out.println("Rs 500");

    }
    public void show()
    {
        System.out.println("This is BookImpl1 class method");
    }
    public void myNewBook()
    {
        System.out.println("new interface method");
    }
    public void dataInfo() {
        System.out.println("Method from other interface");
    }
    public static void main(String[] args) {
        BookImpl4 Book =new BookImpl4();
        Book.bookTitle();
        Book.bookAuthor();
        Book.bookPrice();
        Book.show();
        Book.myNewBook();
        Book.dataInfo();
        }
```

```
<terminated> BookImpl4 [Java Application] C:\Users\harsh
CORE JAVA
ORACLE CROP
Rs 500
This is BookImpl1 class method
new interface method
Method from other interface
```

Class implements Interface Program 2:

BookImpl1.java | BookImpl2.java | NewBook.java | MyNewData.java | BookImpl4.java

```java
package com.evergent.corejava.Interface;

public class BookImpl2 implements Book {
    //When we implements interface in class that class should override all
    //otherwise the class is showing compiletime error
    public void bookTitle()
    {
        System.out.println("CORE JAVA");
    }
    public void bookAuthor()
    {
        System.out.println("ORACLE CROP");
    }
    public void bookPrice()
    {
        System.out.println("Rs 500");

    }
    public void show()
    {
        System.out.println("This is BookImpl1 class method");
    }
    public static void main(String[] args) {
        //we cannot create object to interface but we can create reference
        //we can only access the interface methods if we use the object cr
        //otherwise it shows compile time error
        Book Book =new BookImpl2();
        Book.bookTitle();
        Book.bookAuthor();
        Book.bookPrice();
        //Book.show();
        }
}

```

Outline | Console

```
<terminated> BookImpl2 [Java A
CORE JAVA
ORACLE CROP
Rs 500
```

Class implements Interface Program 3:

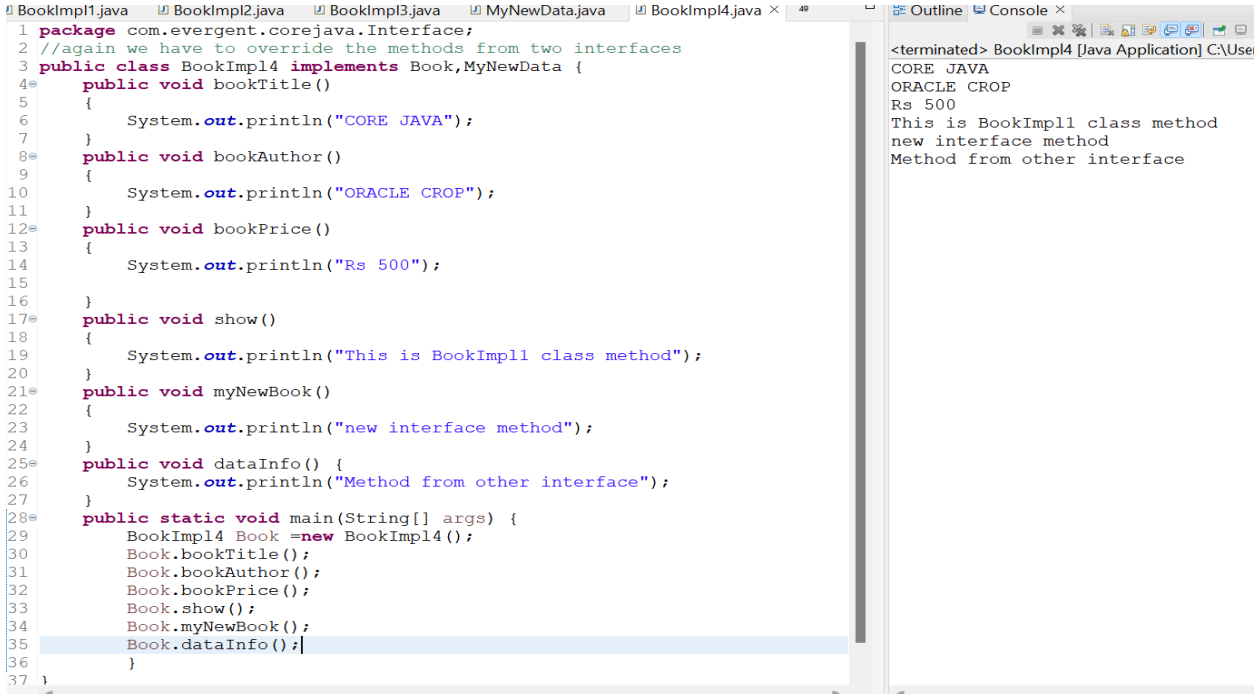
BookImpl1.java | BookImpl2.java | BookImpl3.java | MyNewData.java | BookImpl4.java

```java
package com.evergent.corejava.Interface;
//Java supports multiple inheritance through interfaces
//there is no ambiguity here because both interfaces has only signatures b
//as ambiguity(confusion) there in multiple inheritance using class
//here we have to override all the methods from both interfaces
public class BookImpl3 implements Book,NewBook {
    public void bookTitle()
    {
        System.out.println("CORE JAVA");
    }
    public void bookAuthor()
    {
        System.out.println("ORACLE CROP");
    }
    public void bookPrice()
    {
        System.out.println("Rs 500");

    }
    public void show()
    {
        System.out.println("This is BookImpl1 class method");
    }
    public void myNewBook()
    {
        System.out.println("new interface method");
    }
    public static void main(String[] args) {
        BookImpl3 Book =new BookImpl3();
        Book.bookTitle();
        Book.bookAuthor();
        Book.bookPrice();
        Book.show();
        Book.myNewBook();
        }
}

```

Outline | Console

```
<terminated> BookImpl3 [Java Application] C:\Users\h
CORE JAVA
ORACLE CROP
Rs 500
This is BookImpl1 class method
new interface method
```

Class implements interface program 4:



```java
package com.evergent.corejava.Interface;
//again we have to override the methods from two interfaces
public class BookImpl4 implements Book,MyNewData {
    public void bookTitle()
    {
        System.out.println("CORE JAVA");
    }
    public void bookAuthor()
    {
        System.out.println("ORACLE CROP");
    }
    public void bookPrice()
    {
        System.out.println("Rs 500");

    }
    public void show()
    {
        System.out.println("This is BookImpl1 class method");
    }
    public void myNewBook()
    {
        System.out.println("new interface method");
    }
    public void dataInfo() {
        System.out.println("Method from other interface");
    }
    public static void main(String[] args) {
        BookImpl4 Book =new BookImpl4();
        Book.bookTitle();
        Book.bookAuthor();
        Book.bookPrice();
        Book.show();
        Book.myNewBook();
        Book.dataInfo();
        }
}
```

```
<terminated> BookImpl4 [Java Application] C:\Use
CORE JAVA
ORACLE CROP
Rs 500
This is BookImpl1 class method
new interface method
Method from other interface
```

Date:14-08-2024
Day:8

ABSTRACT CLASS:

1. Abstract class is a **keyword**
2. Abstract class having **abstract methods(unimplemented)** and **concrete(implemented)** methods.
3. If any class having one abstract method that class should be declared as a abstract keyword otherwise the class will be showing **compile time error.**
4. If any class extends abstract class,that class should be **override** all abstract methods,otherwise the class will be showing **compile time error.**
5. We can't create object to abstract class but we can **create reference to abstract class**
6. We can declare abstract class with **zero abstract methods.**

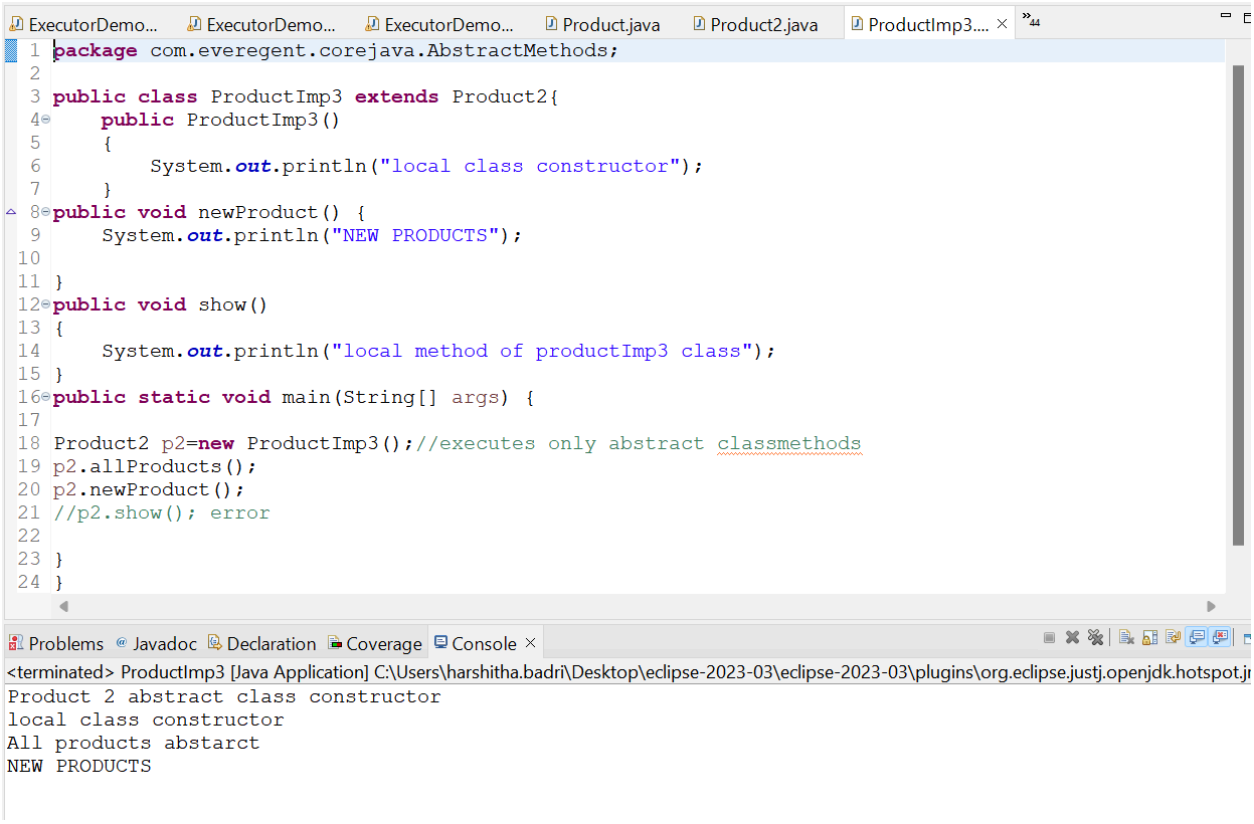
PROGRAM 1:

```java
package com.everegent.corejava.AbstractMethods;
//If any class having one abstract method that class should be
//declared as a abstract keyword otherwise the class
//will be showing compile time error.

public abstract class Product {
    abstract public void newProduct();
    public void allProducts()
    {
        System.out.println("All Products");
    }

}
```

PROGRAM 2:

```java
package com.everegent.corejava.AbstractMethods;

public abstract class Product2 {
    public Product2()
    {
        System.out.println("Product 2 abstract class constructor");
    }
    String name="INDIA";
    abstract public void newProduct();
    public void allProducts()
    {
        System.out.println("All products abstarct");
    }

}
```

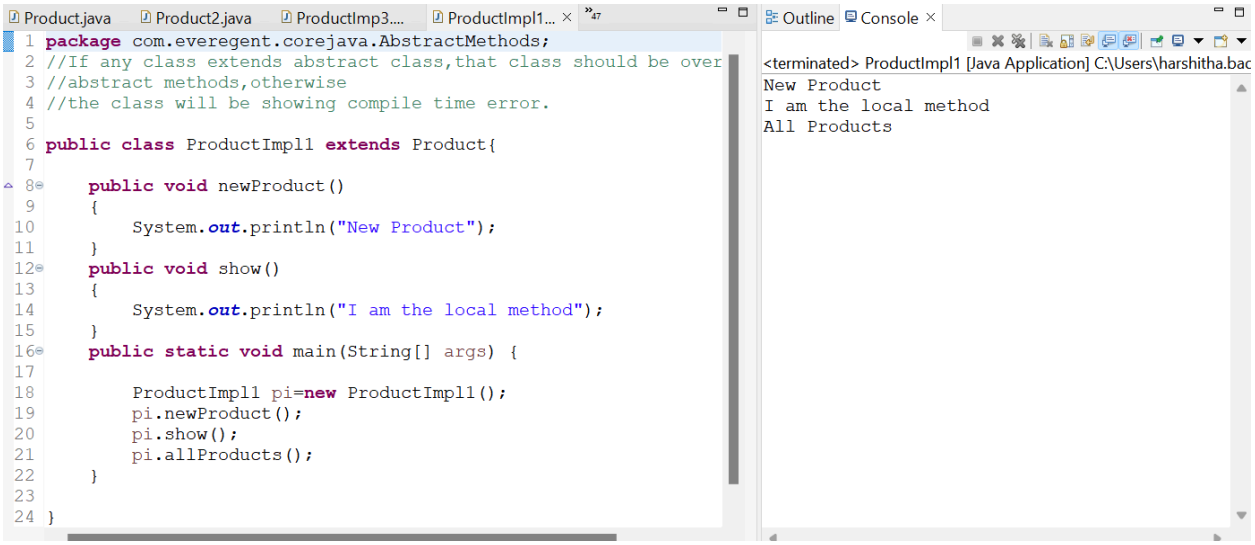
PROGRAM 3:



```
ExecutorDemo... ExecutorDemo... ExecutorDemo... Product.java Product2.java ProductImp3.... ×
 1 package com.everegent.corejava.AbstractMethods;
 2
 3 public class ProductImp3 extends Product2{
 4     public ProductImp3()
 5     {
 6         System.out.println("local class constructor");
 7     }
 8 public void newProduct() {
 9     System.out.println("NEW PRODUCTS");
10
11 }
12 public void show()
13 {
14     System.out.println("local method of productImp3 class");
15 }
16 public static void main(String[] args) {
17
18 Product2 p2=new ProductImp3();//executes only abstract classmethods
19 p2.allProducts();
20 p2.newProduct();
21 //p2.show(); error
22
23 }
24 }
```

```
Problems Javadoc Declaration Coverage Console ×
<terminated> ProductImp3 [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jr
Product 2 abstract class constructor
local class constructor
All products abstarct
NEW PRODUCTS
```

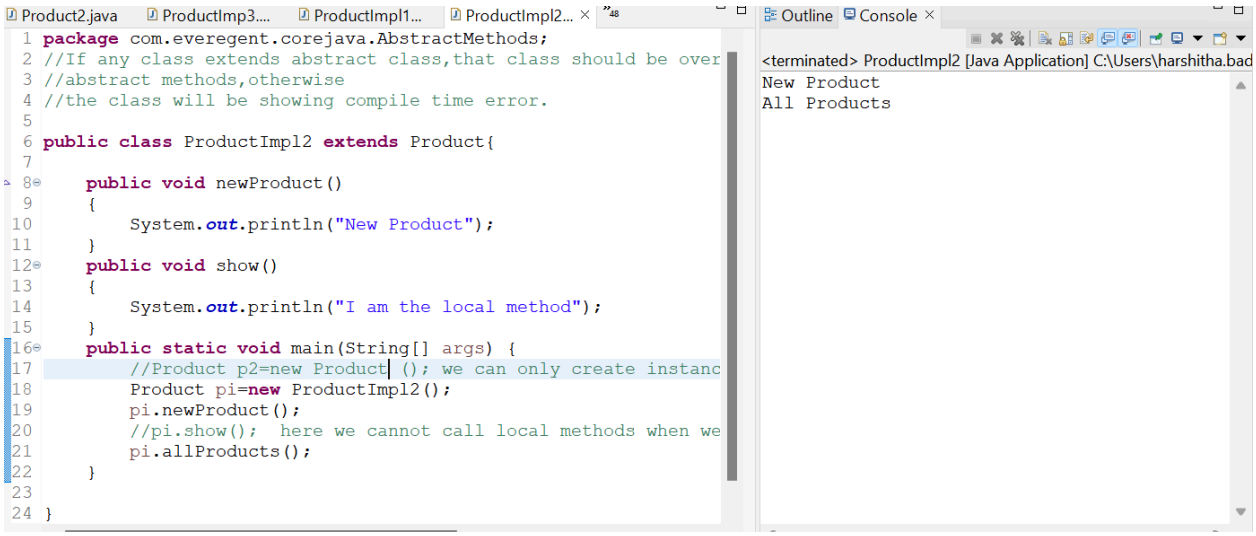
PROGRAM 4:



```
Product.java Product2.java ProductImp3.... ProductImpl1... ×
 1 package com.everegent.corejava.AbstractMethods;
 2 //If any class extends abstract class,that class should be over
 3 //abstract methods,otherwise
 4 //the class will be showing compile time error.
 5
 6 public class ProductImpl1 extends Product{
 7
 8     public void newProduct()
 9     {
10         System.out.println("New Product");
11     }
12     public void show()
13     {
14         System.out.println("I am the local method");
15     }
16     public static void main(String[] args) {
17
18         ProductImpl1 pi=new ProductImpl1();
19         pi.newProduct();
20         pi.show();
21         pi.allProducts();
22     }
23
24 }
```

```
Outline Console ×
<terminated> ProductImpl1 [Java Application] C:\Users\harshitha.bac
New Product
I am the local method
All Products
```

PROGRAM 5:




```java
package com.everegent.corejava.AbstractMethods;
//If any class extends abstract class,that class should be over
//abstract methods,otherwise
//the class will be showing compile time error.

public class ProductImpl2 extends Product{

    public void newProduct()
    {
        System.out.println("New Product");
    }
    public void show()
    {
        System.out.println("I am the local method");
    }
    public static void main(String[] args) {
        //Product p2=new Product (); we can only create instanc
        Product pi=new ProductImpl2();
        pi.newProduct();
        //pi.show();  here we cannot call local methods when we
        pi.allProducts();
    }

}
```

EXCEPTIONAL HANDLING:

1.Exception Handling is mechanism.

2.All Exceptions are executed while abnormal conditions only.

3.Normal flow it won't execute any exceptions.

4.Once any exceptions are occurring in java then remaining lines of code is unreachable.

5.Java.lang.throwable is the supper class for exception and errors.

6.There are two types of exceptions in Java

- Checked exceptions.
- Unchecked exceptions.

7.All checked exceptions are compile time exceptions.

8. All unchecked exceptions are runtime exceptions.

9. There are 5 keywords in exception handling
   a. try{}
   b. catch(){}
   c. finally{}
   d. throws
   e. throw

10. try is for business logic.

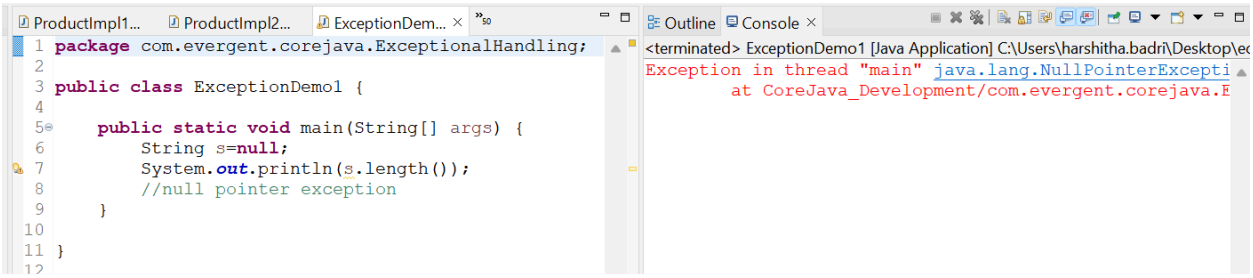
11. catch is for handling exceptions.

12. finally is a block , if exceptions is occurred or not finally block will be execute.

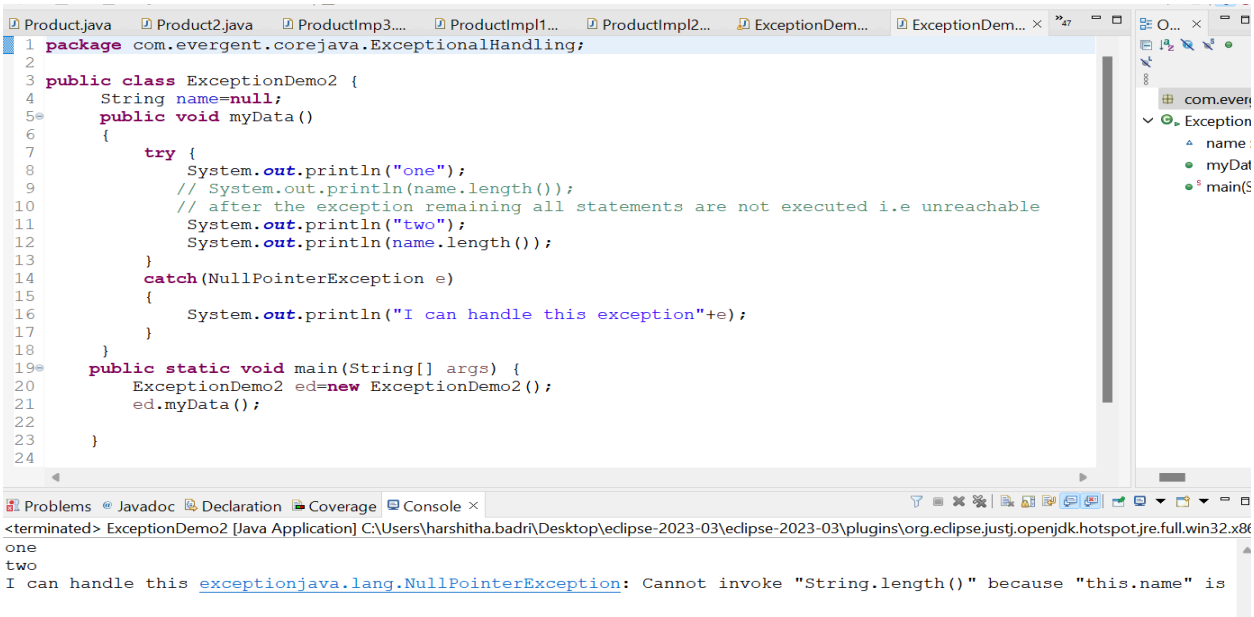
13. throws an exception will be executed method by method.

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.

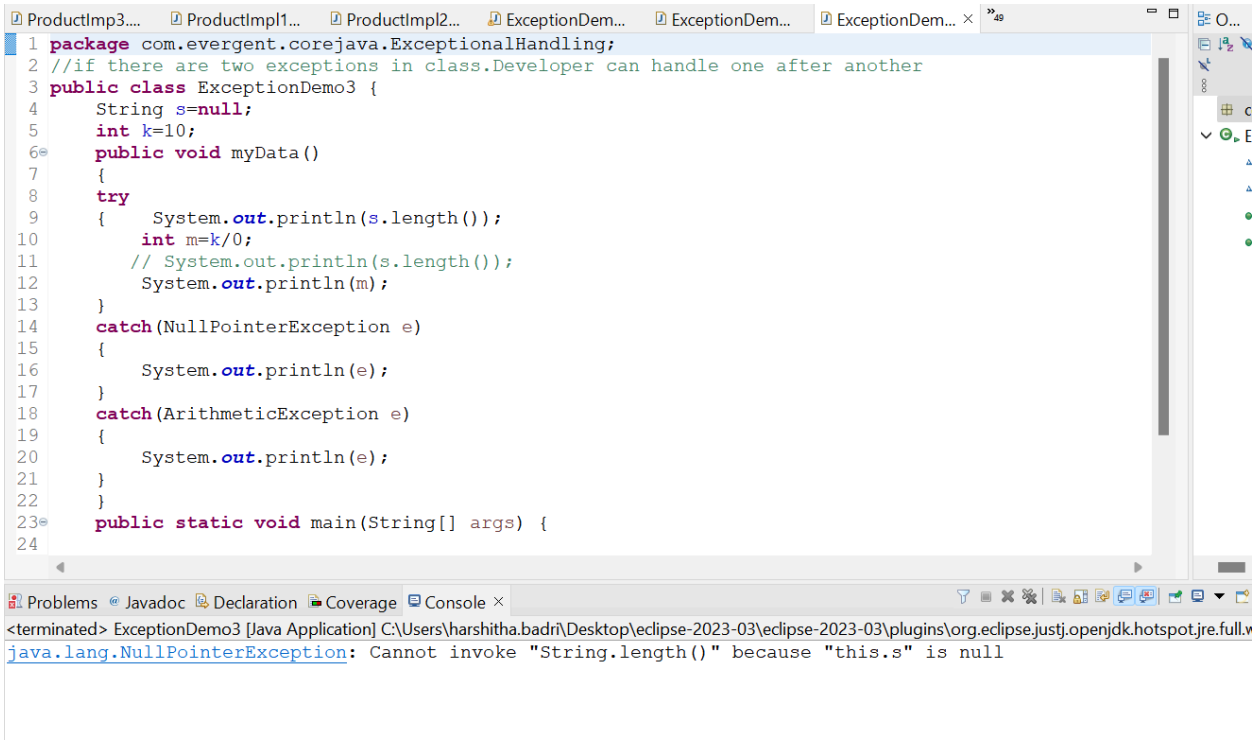
Program 1:



Program 2:

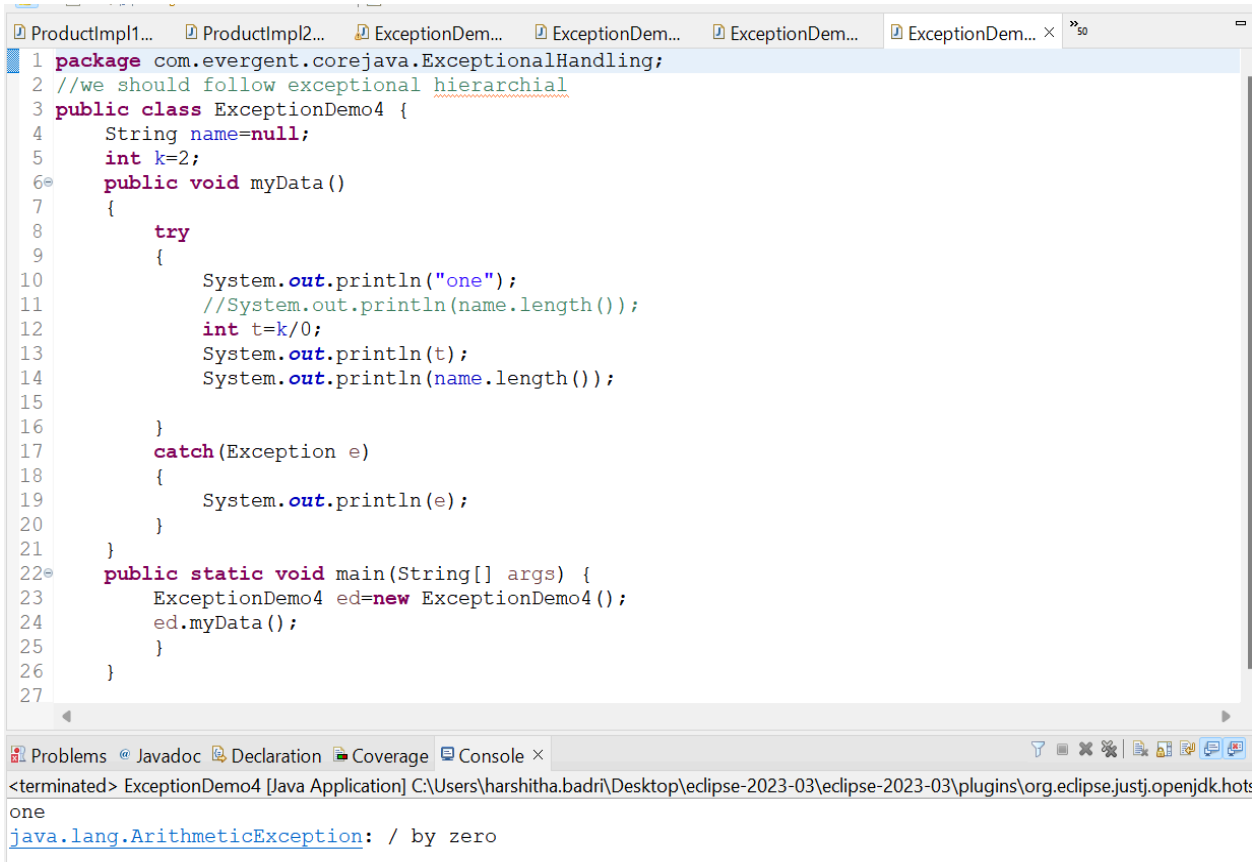


Program 3:



```
ProductImp3....  ProductImpl1...  ProductImpl2...  ExceptionDem...  ExceptionDem...  ExceptionDem...

1  package com.evergent.corejava.ExceptionalHandling;
2  //if there are two exceptions in class.Developer can handle one after another
3  public class ExceptionDemo3 {
4      String s=null;
5      int k=10;
6      public void myData()
7      {
8      try
9      {     System.out.println(s.length());
10         int m=k/0;
11        // System.out.println(s.length());
12         System.out.println(m);
13     }
14     catch(NullPointerException e)
15     {
16         System.out.println(e);
17     }
18     catch(ArithmeticException e)
19     {
20         System.out.println(e);
21     }
22     }
23     public static void main(String[] args) {
24

Problems  Javadoc  Declaration  Coverage  Console

<terminated> ExceptionDemo3 [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.w
java.lang.NullPointerException: Cannot invoke "String.length()" because "this.s" is null
```

Program 4:



```
ProductImpl1...  ProductImpl2...  ExceptionDem...  ExceptionDem...  ExceptionDem...  ExceptionDem...

1  package com.evergent.corejava.ExceptionalHandling;
2  //we should follow exceptional hierarchial
3  public class ExceptionDemo4 {
4      String name=null;
5      int k=2;
6      public void myData()
7      {
8          try
9          {
10             System.out.println("one");
11             //System.out.println(name.length());
12             int t=k/0;
13             System.out.println(t);
14             System.out.println(name.length());
15
16         }
17         catch(Exception e)
18         {
19             System.out.println(e);
20         }
21     }
22     public static void main(String[] args) {
23         ExceptionDemo4 ed=new ExceptionDemo4();
24         ed.myData();
25         }
26     }
27

Problems  Javadoc  Declaration  Coverage  Console

<terminated> ExceptionDemo4 [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hots
one
java.lang.ArithmeticException: / by zero
```

Program 5:

```java
package com.evergent.corejava.ExceptionalHandling;
//finally is a block if exception occurs or not finally block will be executed
public class ExceptionDemo5 {
   public static void main(String[] args) {
        try
        {
            System.out.println("one");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        finally
        {
            System.out.println("I am finally block");
        }
    }

}
```

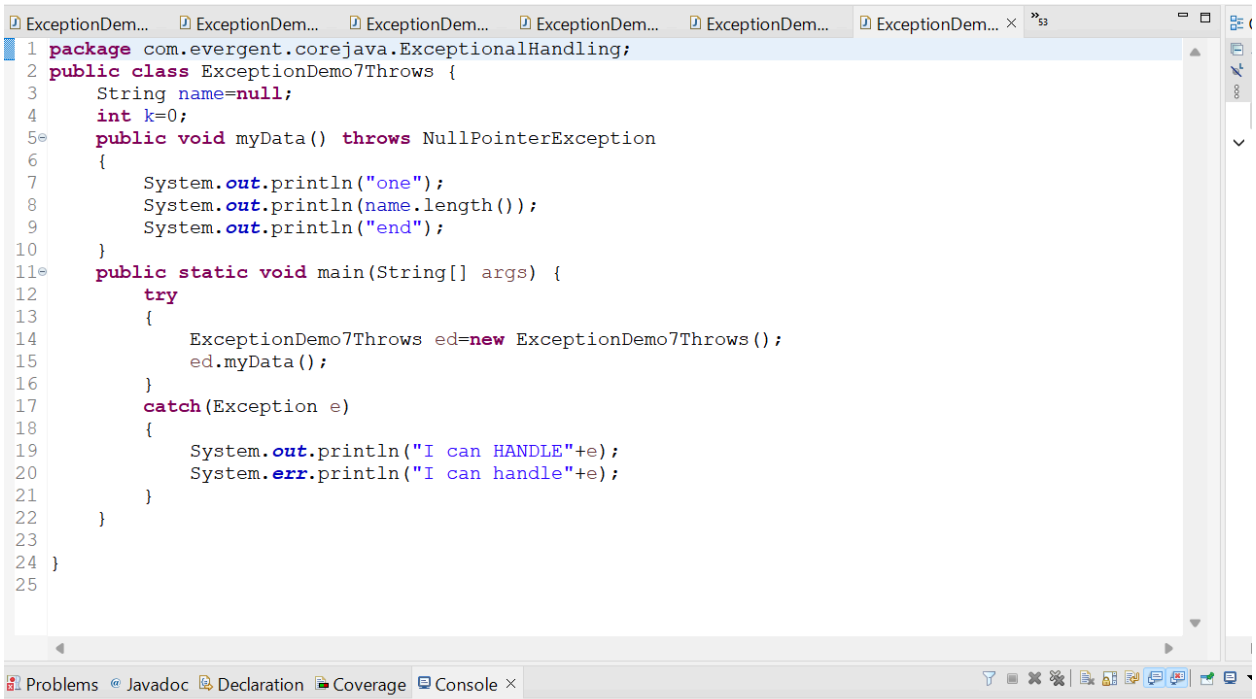
```
<terminated> ExceptionDemo5 [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk
one
I am finally block
```

Program 6:

```java
package com.evergent.corejava.ExceptionalHandling;
//try followed by either catch or finally block
public class ExceptionDemo6 {
    String name=null;
    int k=2;
    public void myData()
    {
        try
        {
            System.out.println("me");
            System.out.println(name.length());
            int t=k/0;
            System.out.println(t);
        }
        finally
        {
            System.out.println("I am finally block");
        }
    }
    public static void main(String[] args) {

        ExceptionDemo6 ed=new ExceptionDemo6();
        ed.myData();
    }

}
```

```
<terminated> ExceptionDemo6 [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86
me
I am finally block
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" because "this.name" is
        at CoreJava_Development/com.evergent.corejava.ExceptionalHandling.ExceptionDemo6.myData(ExceptionDemo6.jav
        at CoreJava_Development/com.evergent.corejava.ExceptionalHandling.ExceptionDemo6.main(ExceptionDemo6.java:
```

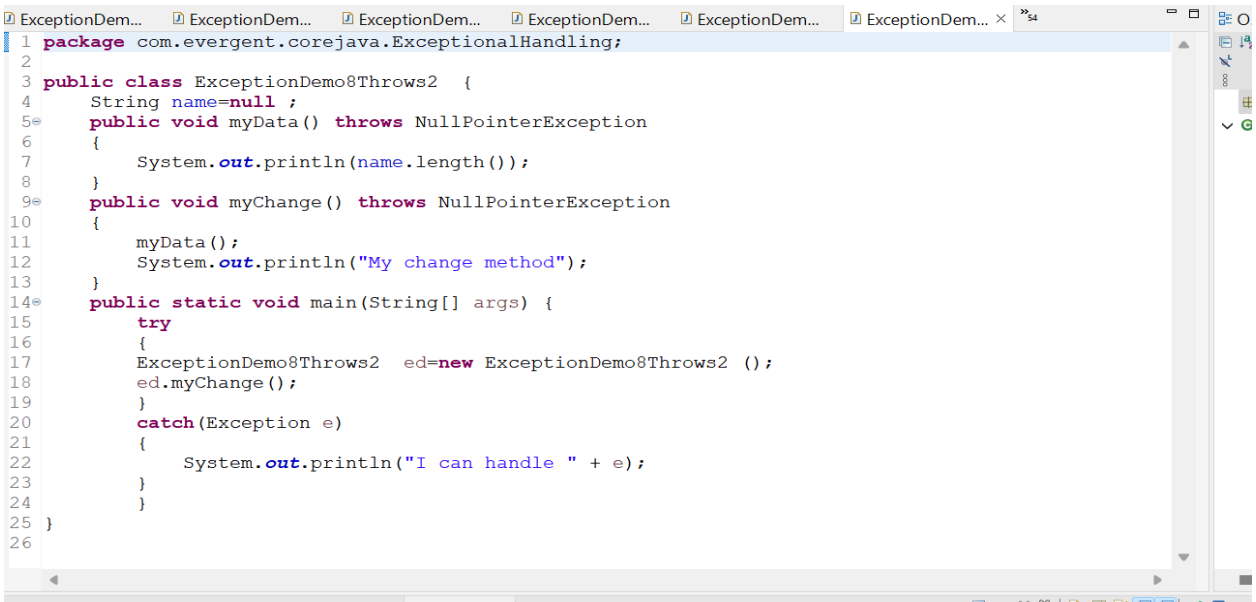
Program 7:



```
package com.evergent.corejava.ExceptionalHandling;
public class ExceptionDemo7Throws {
    String name=null;
    int k=0;
    public void myData() throws NullPointerException
    {
        System.out.println("one");
        System.out.println(name.length());
        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);
            System.err.println("I can handle"+e);
        }
    }

}

```

```
<terminated> ExceptionDemo7Throws [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hots
one
I can handlejava.lang.NullPointerException: Cannot invoke "String.length()" because "this.name" is null
I can HANDLEjava.lang.NullPointerException: Cannot invoke "String.length()" because "this.name" is null
```

Program 8:



```
package com.evergent.corejava.ExceptionalHandling;

public class ExceptionDemo8Throws2  {
    String name=null ;
    public void myData() throws NullPointerException
    {
        System.out.println(name.length());
    }
    public void myChange() throws NullPointerException
    {
        myData();
        System.out.println("My change method");
    }
    public static void main(String[] args) {
        try
        {
        ExceptionDemo8Throws2  ed=new ExceptionDemo8Throws2 ();
        ed.myChange();
        }
        catch(Exception e)
        {
            System.out.println("I can handle " + e);
        }
        }
}

```

```
terminated> ExceptionDemo7Throws [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotsp
one
I can handlejava.lang.NullPointerException: Cannot invoke "String.length()" because "this.name" is null
I can HANDLEjava.lang.NullPointerException: Cannot invoke "String.length()" because "this.name" is null
```

Program 9:

```java
package com.evergent.corejava.ExceptionalHandling;
import java.io.File;
public class FileNotFound {

    public static void main(String[] args) {
        try
        {
            File f=new File("C:/Users/zharshitha.badri/1723629754945_CoreJAVA-ExamPaper (1).txt")
            Scanner sc=new Scanner(f);
            while(sc.hasNextLine())
            {
                System.out.println(sc.nextLine());
            }
            sc.close();
        }
        catch(FileNotFoundException e)
        {
            e.printStackTrace();
        }


    }

}
```

Problems Javadoc Declaration Coverage Console

```
erminated> FileNotFound [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_6
ava.io.FileNotFoundException: C:\Users\zharshitha.badri\1723629754945_CoreJAVA-ExamPaper (1).txt (The system cann
        at java.base/java.io.FileInputStream.open0(Native Method)
        at java.base/java.io.FileInputStream.open(FileInputStream.java:216)
        at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
        at java.base/java.util.Scanner.<init>(Scanner.java:639)
        at CoreJava_Development/com.evergent.corejava.ExceptionalHandling.FileNotFound.main(FileNotFound.java:11)
```

Program 10:

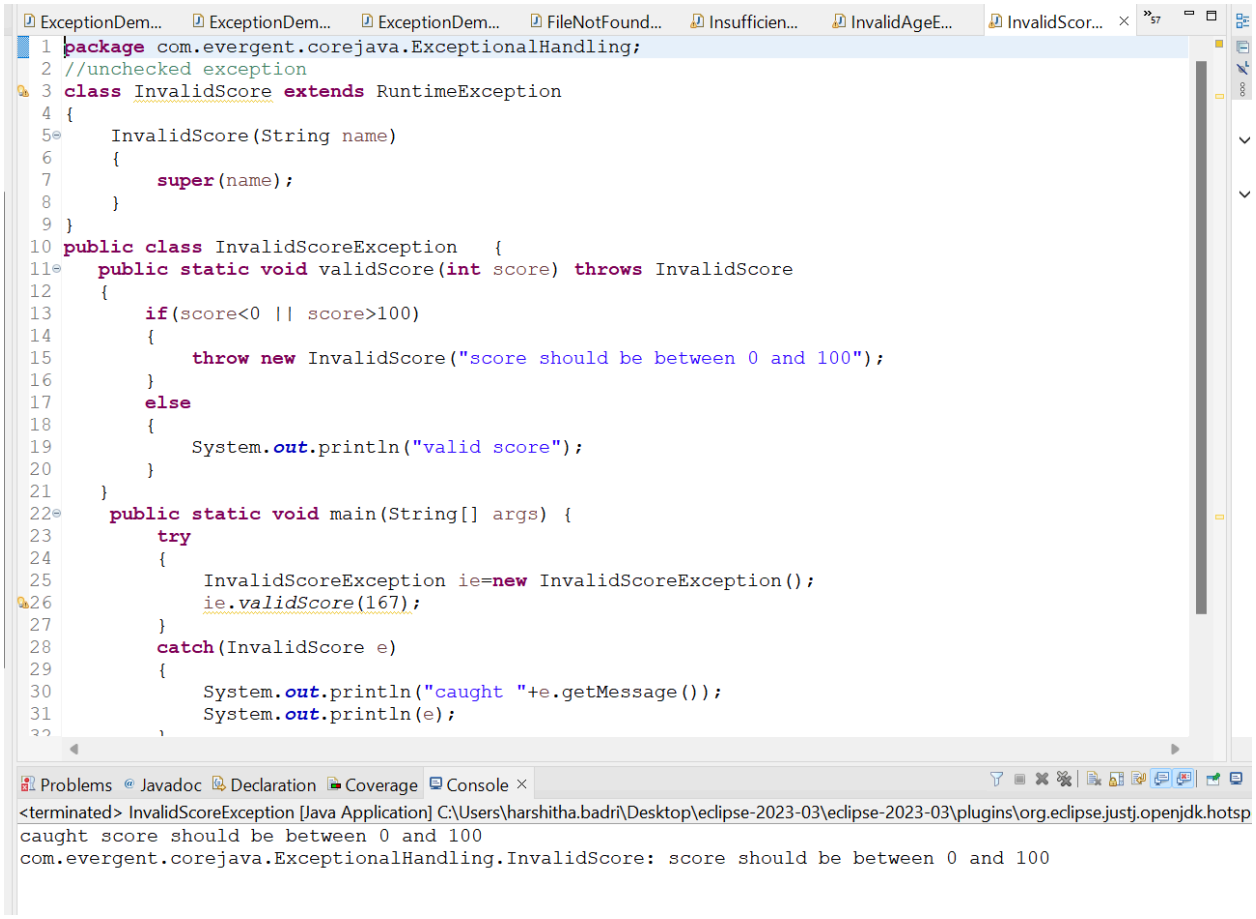
ExceptionDem... ExceptionDem... ExceptionDem... ExceptionDem... FileNotFound... Insufficien...

```java
package com.evergent.corejava.ExceptionalHandling;
class InsufficientFunds extends Exception
{
    public InsufficientFunds(String msg)
    {
        super(msg);
    }
}
public class InsufficientFundsException {
    public static void withdraw(double amount) throws InsufficientFunds
    {
        double balance=50000;
        if(amount > balance)
        throw new InsufficientFunds("InsufficientFunds for withdrawl");
        else
        System.out.println("withdraw successfull");
    }
    public static void main(String[] args) {
        try
        {
        InsufficientFundsException ife=new InsufficientFundsException();
        ife.withdraw(56000);
        }
        catch(InsufficientFunds e)
        {
            System.out.println("caught exception "+e.getMessage());
            System.out.println(e);
        }
        System.out.println("pgm continues after handling");

    }
```

Problems Javadoc Declaration Coverage Console

```
<terminated> InsufficientFundsException [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.c
caught exception InsufficientFunds for withdrawl
com.evergent.corejava.ExceptionalHandling.InsufficientFunds: InsufficientFunds for withdrawl
pgm continues after handling
```

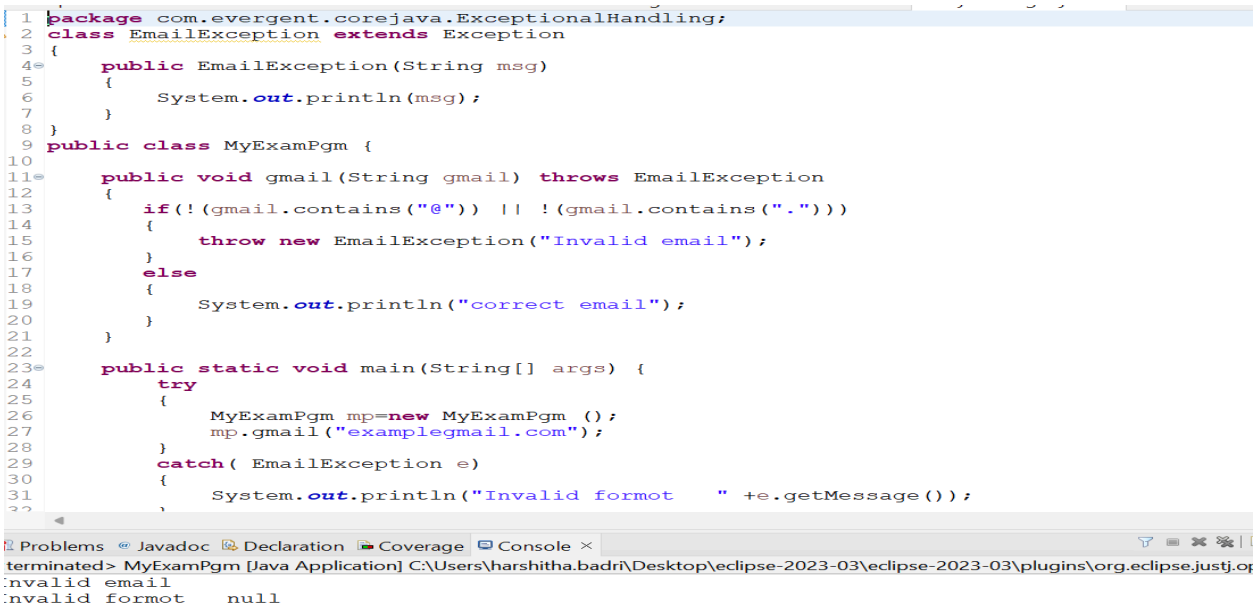
Program 11:



```
package com.evergent.corejava.ExceptionalHandling;
//unchecked exception
class InvalidScore extends RuntimeException
{
    InvalidScore(String name)
    {
        super(name);
    }
}
public class InvalidScoreException   {
   public static void validScore(int score) throws InvalidScore
   {
       if(score<0 || score>100)
       {
           throw new InvalidScore("score should be between 0 and 100");
       }
       else
       {
           System.out.println("valid score");
       }
   }
    public static void main(String[] args) {
        try
        {
            InvalidScoreException ie=new InvalidScoreException();
            ie.validScore(167);
        }
        catch(InvalidScore e)
        {
            System.out.println("caught "+e.getMessage());
            System.out.println(e);
```

```
<terminated> InvalidScoreException [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotsp
caught score should be between 0 and 100
com.evergent.corejava.ExceptionalHandling.InvalidScore: score should be between 0 and 100
```

Program 12:



```
package com.evergent.corejava.ExceptionalHandling;
class EmailException extends Exception
{
    public EmailException(String msg)
    {
        System.out.println(msg);
    }
}
public class MyExamPgm {

    public void gmail(String gmail) throws EmailException
    {
        if(!(gmail.contains("@")) || !(gmail.contains(".")))
        {
            throw new EmailException("Invalid email");
        }
        else
        {
            System.out.println("correct email");
        }
    }

    public static void main(String[] args) {
        try
        {
            MyExamPgm mp=new MyExamPgm ();
            mp.gmail("examplegmail.com");
        }
        catch( EmailException e)
        {
            System.out.println("Invalid formot    " +e.getMessage());
```

```
terminated> MyExamPgm [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-2023-03\plugins\org.eclipse.justj.op
nvalid email
nvalid formot    null
```

Program 13:

```java
package com.evergent.corejava.ExceptionalHandling;

class ProductNotFound extends Exception {
    public ProductNotFound(String message)
    {
        System.out.println("Hello" +message);
    }
}
public class ProductNotFoundException
{
    int pno=160;
    public void myData() throws ProductNotFound
    {
        if(pno>150)
        {
        throw new ProductNotFound("There are no products");
        }
        else
        System.out.println("pRoduct found");
    }

public static void main(String[] args) {
    try
    {
        ProductNotFoundException p=new ProductNotFoundException();
        p.myData();
    }
    catch(Exception e)
    {
        System.out.println("handled  "+e);
    }
```

Problems Javadoc Declaration Coverage Console ×

```
terminated> ProductNotFoundException [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-0
HelloThere are no products
handled  com.evergent.corejava.ExceptionalHandling.ProductNotFound
```

Program 14:

```java
package com.evergent.corejava.ExceptionalHandling;

public class ArrayIndexOutOfTheBoxException13 {

    public static void main(String[] args) {
        int[] num= {1,4,6,8,3};
        try
        {
            System.out.println("accessing elmt"+num[8]);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("caught "+e);
        }

    }

}
```

Problems Javadoc Declaration Coverage Console

```
<terminated> ArrayIndexOutOfTheBoxException13 [Java Application] C:\Users\harshitha.badri\Desktop\eclipse-2023-03\eclipse-20
caught java.lang.ArrayIndexOutOfBoundsException: Index 8 out of bounds for length 5
```

JAVA BEANS:

Program 1:

ProductNotFo... commandLineA... ArrayIndexO... EmployeeImp....

```java
package com.evergent.corejava.javabeans;
import java.io.Serializable;
 class Employee implements Serializable
{
    private int eno;
    private String ename;
    private double sal;
    public void setEno(int eno)
    {
        this.eno=eno;
    }
    public int getEno()
    {
        return eno;
    }
    public void setEname(String ename)
    {
        this.ename=ename;
    }
    public String getEname()
    {
        return ename;
    }
    public void setsal(double sal)
    {
        this.sal=sal;
    }
    public double getsal()
    {
        return sal;
    }
}
public class EmployeeImp {

    public static void main(String[] args) {
         Employee e=new  Employee ();
         e.setEno(500);
```

Console Outline

```
<terminated> EmployeeImp [Java Application] C:\Users\harshitha.b
emp no. 500
emp name har
emp no. 50000.0
```

Program 2:

```java
package com.evergent.corejava.javabeans;
import java.io.Serializable;
 class product implements Serializable
{
    private int pno;
    private String pname;
    private double price;
    public product(int pno,String pname,double price)
    {
        this.pno=pno;
        this.pname=pname;
        this.price=price;
    }

    public int getPno()
    {
        return pno;
    }

    public String getPname()
    {
        return pname;
    }

    public double getPrice()
    {
        return price;
    }
}
public class ProductImp {

    public static void main(String[] args) {
         product e=new  product (500,"item",600.0);
         System.out.println("product no. "+e.getPno());
         System.out.println("product name "+e.getPname());
         System.out.println("product price. "+e.getPrice());

```

```
<terminated> ProductImp [Java Applicatio
product no. 500
product name item
product price. 600.0
```

Program 3:

```java
package com.evergent.corejava.javabeans;

import java.io.Serializable;

public class StudentBean implements Serializable {
    private String sname;
    private int sno;
    private String address;
    public void setsname(String sname)
    {
        this.sname=sname;
    }
    public void setsno(int sno)
    {
        this.sno=sno;
    }
    public void setAddress(String address)
    {
        this.address=address;
    }
    public String toString()
    {
        return "student no: " +sno+ "\n student name : "+sname+

    }
    public static void main(String[] args) {
        StudentBean sb=new StudentBean();
        sb.setsno(100);
        sb.setsname("sai");
        sb.setAddress("metpally");
        System.out.println(sb);
    }

```

```
<terminated> StudentBean [Java Application] C:\
student no: 100
 student name : sai
 student address metpally
```