INDEX

DAY-1 (05/08/2024)

DAY-2 (06/08/2024)

DAY-3 (07/08/2024)

OOPS

Encapsulation

- Calculation
- Person class
- MethodFlow

3. About System.out.println()
4. Inheritance
5. Polymorphism
   - Overloading
   - Overriding
6. Abstract
7. IS-A-Realationship(Inheritance)
8. HAS-A-Realationship(Inheritance)

DAY-4 (08/08/2024)

Constructor

Class name and constructor name should be same

There are 2 types of constructors

- Default Constructor
- Parameterized Constructor

iii. We can access constructor while creation of object

iv. Constructors are mainly for intializing instance variables

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 deafult constructor

vii. this, super are keywords

This :-

--> this is a keyword always refers to instance variables (ex:- this.emp)

--> this is also used to call methods of current class (this.show())

--> this is also used to call constructors of current class inside a constructor(this() or this(para);)

Super :-

? super is a keyword always refers to instance variables of super class (ex:- super.emp)

? super is also used to call methods of super class (ex:- super.show())

? super is also used to call constructors of super class inside a constructor of current class (ex:- super() or super(para);)

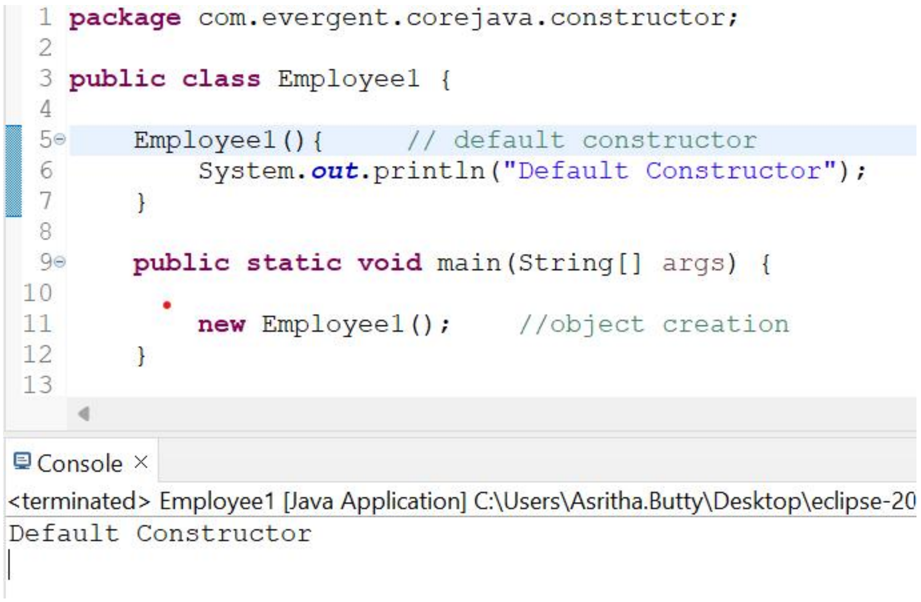
? **Super() is defaulty called in every constructor of its subclass that why we should always keepa default const in a parent class when it's chils class has a construtor**

? Only this() or super() or anyone of them having parameters can be called in a child constructor. Note:- only one and that shouls be in the 1st line

viii. Always constructor are overloaded

ix. Copy constructor

Points covered in this Picture : 1,2,3



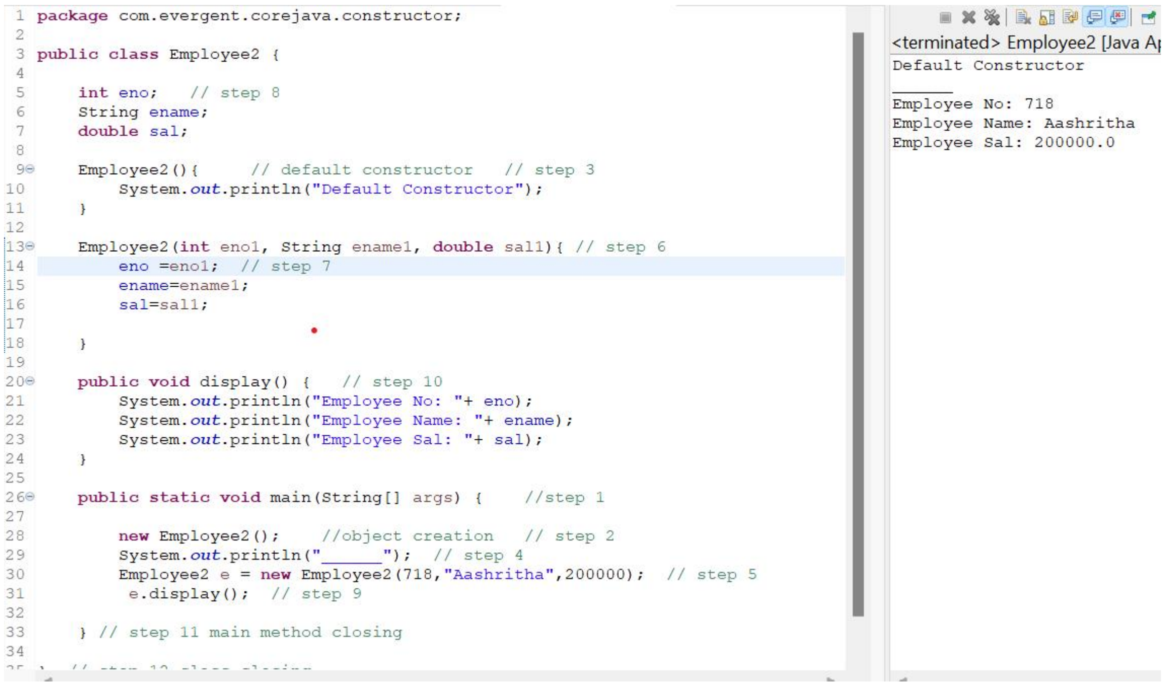
```
package com.evergent.corejava.constructor;

public class Employee1 {

    Employee1(){      // default constructor
        System.out.println("Default Constructor");
    }

    public static void main(String[] args) {

        new Employee1();     //object creation
    }

```

Console ×

<terminated> Employee1 [Java Application] C:\Users\Asritha.Butty\Desktop\eclipse-20

```
Default Constructor
```

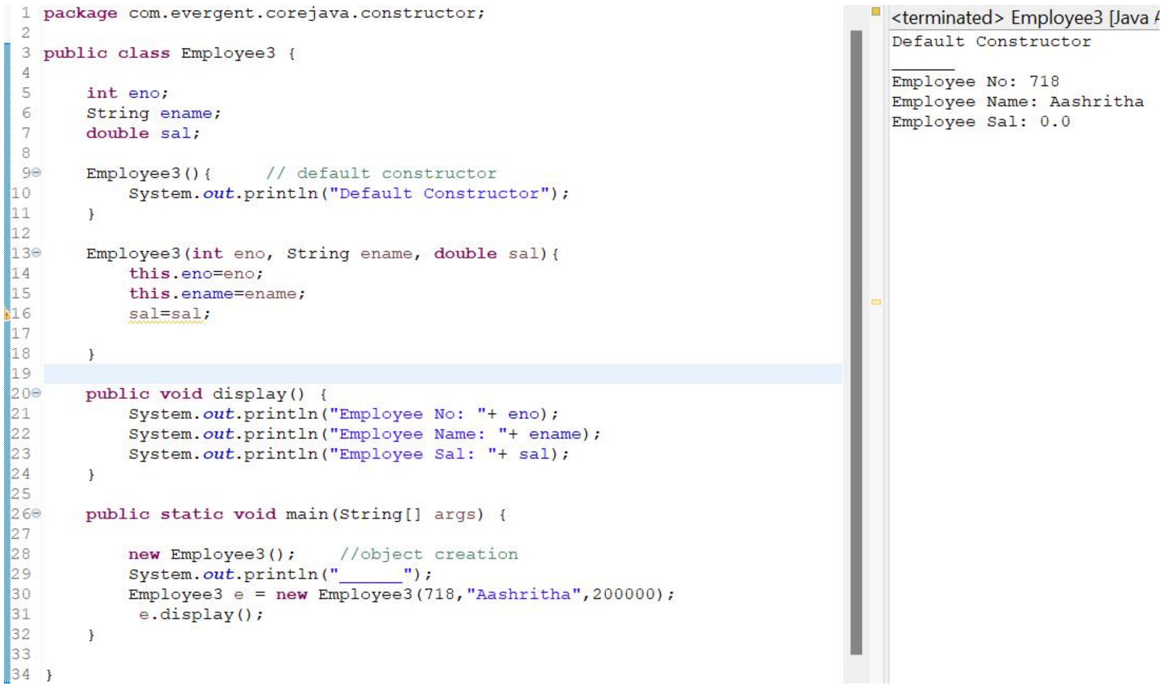
2. **Points covered are: 2,3,4**



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

public class Employee2 {

    int eno;   // step 8
    String ename;
    double sal;

    Employee2(){     // default constructor   // step 3
        System.out.println("Default Constructor");
    }

    Employee2(int eno1, String ename1, double sal1){ // step 6
        eno =eno1;  // step 7
        ename=ename1;
        sal=sal1;

    }

    public void display() {   // step 10
        System.out.println("Employee No: "+ eno);
        System.out.println("Employee Name: "+ ename);
        System.out.println("Employee Sal: "+ sal);
    }

    public static void main(String[] args) {    //step 1

        new Employee2();   //object creation   // step 2
        System.out.println("______");  // step 4
        Employee2 e = new Employee2(718,"Aashritha",200000);  // step 5
         e.display();  // step 9

    } // step 11 main method closing

}  // step 12 class closing
```

```
<terminated> Employee2 [Java Ap
Default Constructor
______
Employee No: 718
Employee Name: Aashritha
Employee Sal: 200000.0
```

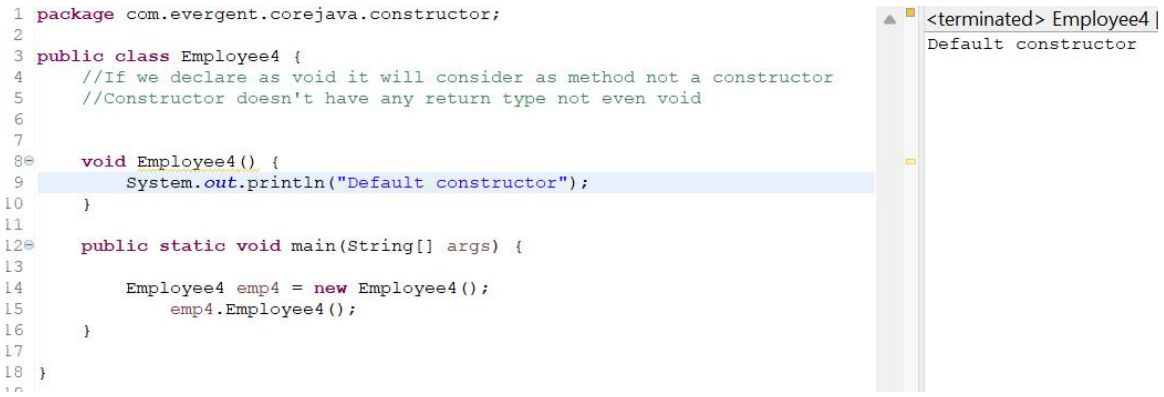
3. **Points covered are: 4,7(this keyword),8**



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

public class Employee3 {

    int eno;
    String ename;
    double sal;

    Employee3(){     // default constructor
        System.out.println("Default Constructor");
    }

    Employee3(int eno, String ename, double sal){
        this.eno=eno;
        this.ename=ename;
        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) {

        new Employee3();    //object creation
        System.out.println("______");
        Employee3 e = new Employee3(718,"Aashritha",200000);
         e.display();
    }

}
```

```
<terminated> Employee3 [Java /
Default Constructor
______
Employee No: 718
Employee Name: Aashritha
Employee Sal: 0.0
```

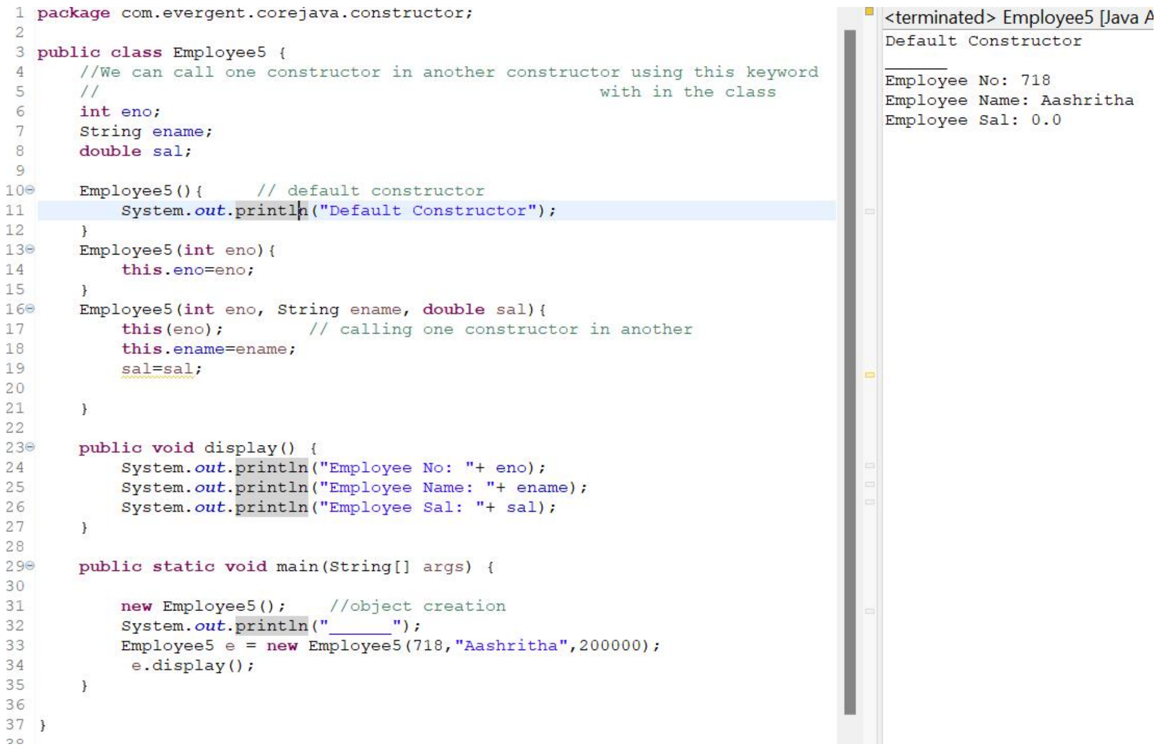
4. **Points Covered are: 5**



```
package com.evergent.corejava.constructor;

public class Employee4 {
    //If we declare as void it will consider as method not a constructor
    //Constructor doesn't have any return type not even void


    void Employee4() {
        System.out.println("Default constructor");
    }

    public static void main(String[] args) {

        Employee4 emp4 = new Employee4();
            emp4.Employee4();
    }

}
```

```
<terminated> Employee4 |
Default constructor
```

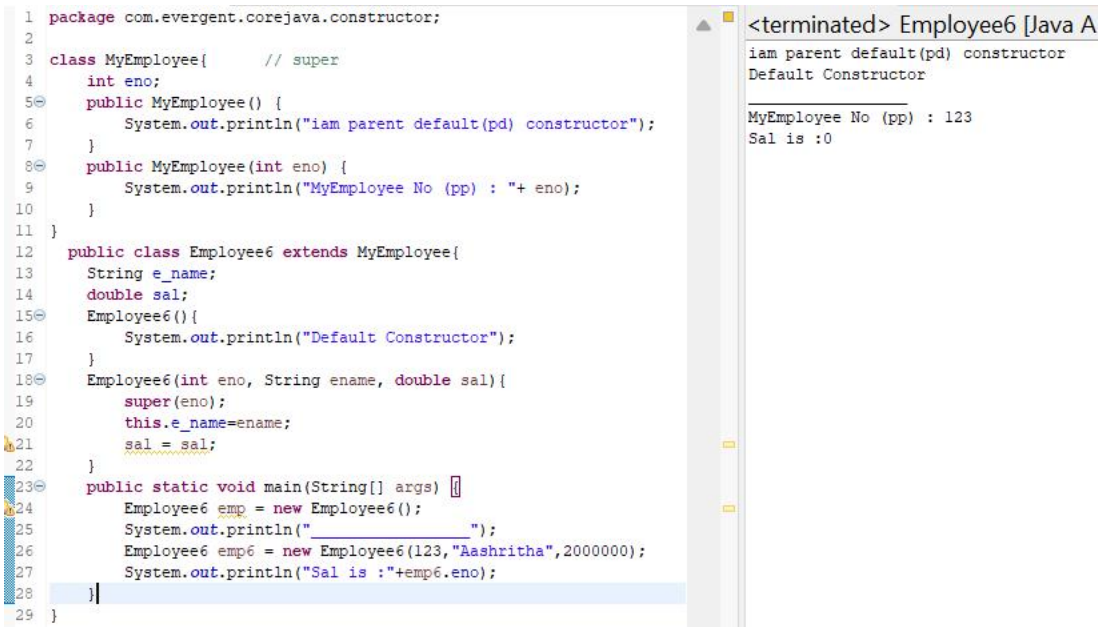
5. **Points covered are: 7**



```
package com.evergent.corejava.constructor;

public class Employee5 {
    //We can call one constructor in another constructor using this keyword
    //                                              with in the class
    int eno;
    String ename;
    double sal;

    Employee5(){      // default constructor
        System.out.println("Default Constructor");
    }
    Employee5(int eno){
        this.eno=eno;
    }
    Employee5(int eno, String ename, double sal){
        this(eno);        // calling one constructor in another
        this.ename=ename;
        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) {

        new Employee5();    //object creation
        System.out.println("______");
        Employee5 e = new Employee5(718,"Aashritha",200000);
         e.display();
    }

}
```

```
<terminated> Employee5 [Java A
Default Constructor
______
Employee No: 718
Employee Name: Aashritha
Employee Sal: 0.0
```

6. **Points covered are: 7**

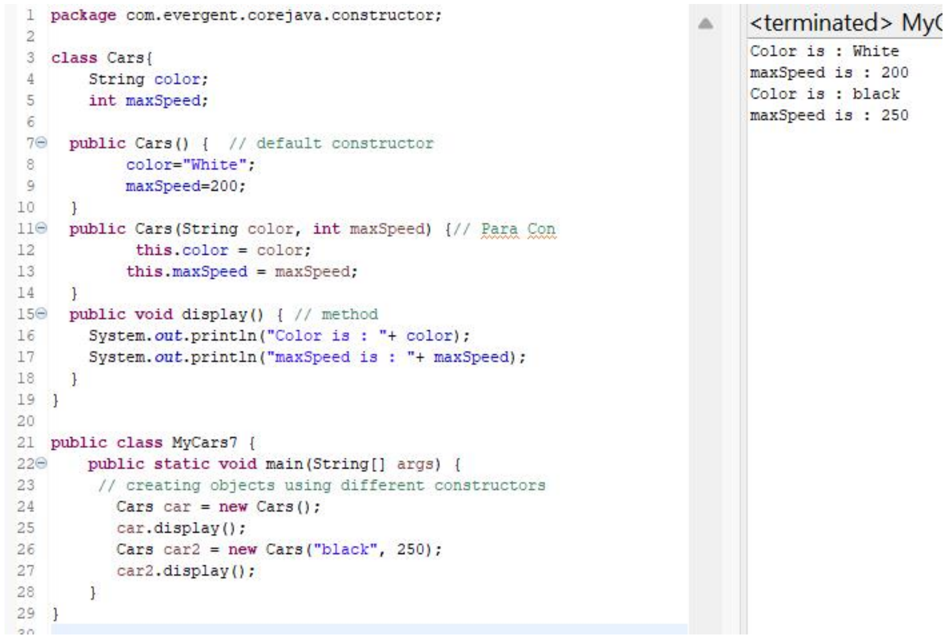


```
package com.evergent.corejava.constructor;

class MyEmployee{       // super
    int eno;
    public MyEmployee() {
        System.out.println("iam parent default(pd) constructor");
    }
    public MyEmployee(int eno) {
        System.out.println("MyEmployee No (pp) : "+ eno);
    }
}
  public class Employee6 extends MyEmployee{
    String e_name;
    double sal;
    Employee6(){
        System.out.println("Default Constructor");
    }
    Employee6(int eno, String ename, double sal){
        super(eno);
        this.e_name=ename;
        sal = sal;
    }
    public static void main(String[] args) {
        Employee6 emp = new Employee6();
        System.out.println("________________");
        Employee6 emp6 = new Employee6(123,"Aashritha",2000000);
        System.out.println("Sal is :"+emp6.eno);
    }
}
```

<terminated> Employee6 [Java A

```
iam parent default(pd) constructor
Default Constructor
________________
MyEmployee No (pp) : 123
Sal is :0
```

7. **Points covered are: 8**

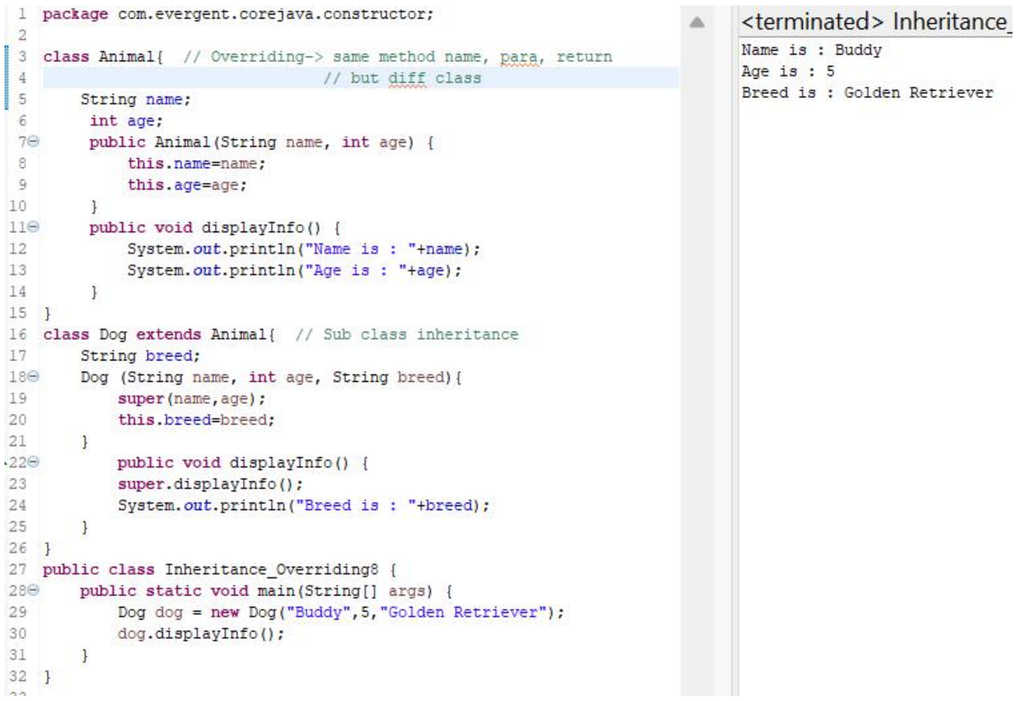


```
package com.evergent.corejava.constructor;

class Cars{
    String color;
    int maxSpeed;

  public Cars() {  // default constructor
        color="White";
        maxSpeed=200;
  }
  public Cars(String color, int maxSpeed) {// Para Con
         this.color = color;
        this.maxSpeed = maxSpeed;
  }
  public void display() { // method
    System.out.println("Color is : "+ color);
    System.out.println("maxSpeed is : "+ maxSpeed);
  }
}

public class MyCars7 {
    public static void main(String[] args) {
     // creating objects using different constructors
       Cars car = new Cars();
       car.display();
       Cars car2 = new Cars("black", 250);
       car2.display();
    }
}
```

<terminated> My(

```
Color is : White
maxSpeed is : 200
Color is : black
maxSpeed is : 250
```

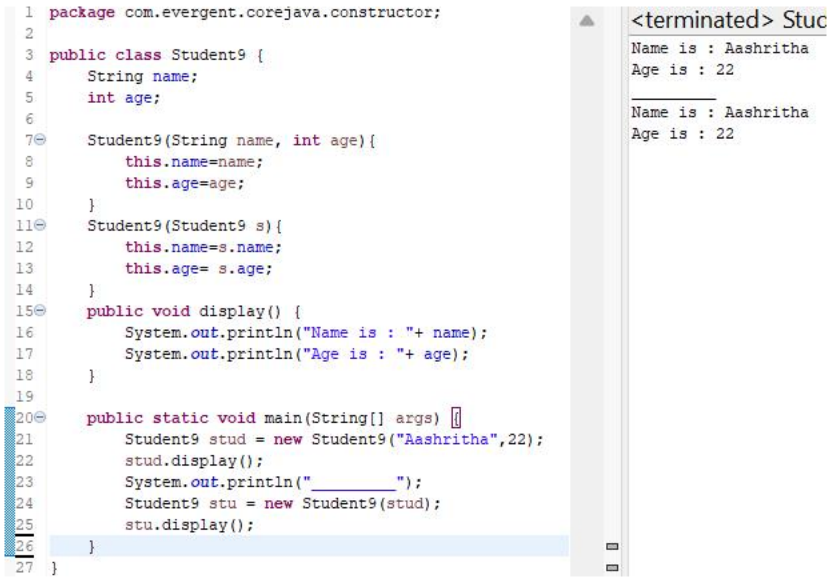
8. **Points covered are: 8 [Inheritance_Overriding]**



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

class Animal{  // Overriding-> same method name, para, return
                              // but diff class
    String name;
     int age;
     public Animal(String name, int age) {
         this.name=name;
         this.age=age;
     }
     public void displayInfo() {
         System.out.println("Name is : "+name);
         System.out.println("Age is : "+age);
     }
}
class Dog extends Animal{  // Sub class inheritance
    String breed;
    Dog (String name, int age, String breed){
        super(name,age);
        this.breed=breed;
    }
        public void displayInfo() {
        super.displayInfo();
        System.out.println("Breed is : "+breed);
    }
}
public class Inheritance_Overriding8 {
    public static void main(String[] args) {
        Dog dog = new Dog("Buddy",5,"Golden Retriever");
        dog.displayInfo();
    }
}
```

```
<terminated> Inheritance_
Name is : Buddy
Age is : 5
Breed is : Golden Retriever
```

9. **Points covered are: 9 Copy Constructor**



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

public class Student9 {
    String name;
    int age;

    Student9(String name, int age){
        this.name=name;
        this.age=age;
    }
    Student9(Student9 s){
        this.name=s.name;
        this.age= s.age;
    }
    public void display() {
        System.out.println("Name is : "+ name);
        System.out.println("Age is : "+ age);
    }

    public static void main(String[] args) {
        Student9 stud = new Student9("Aashritha",22);
        stud.display();
        System.out.println("_________");
        Student9 stu = new Student9(stud);
        stu.display();
    }
}
```

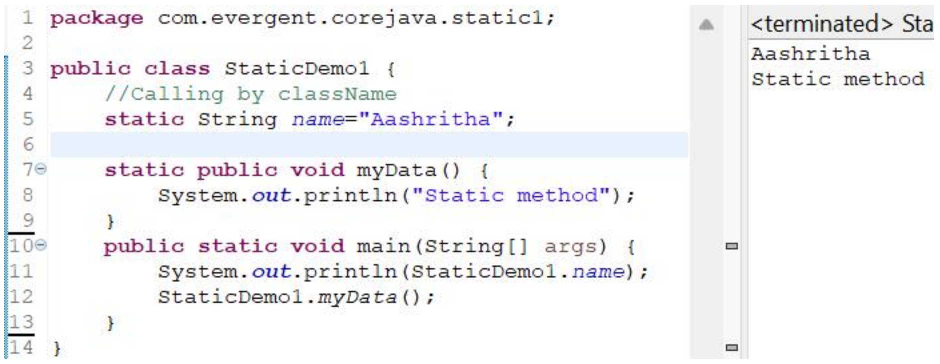
```
<terminated> Stuc
Name is : Aashritha
Age is : 22
_________
Name is : Aashritha
Age is : 22
```

DAY-5 (09/08/2024)

Static :-

Static is a keyword
We can declare static as variables & Methods
We can access static variables and methods direct through classname.methodname and classname.variablename.
Static methods can access static methods and static variables only
Static methods cannot access non static methods and non static variables
Non static methods can access static methods and static variables
Static Block ---> Whenever class is loaded into JVM at that time static blocks are intialised first.
Static Block ---> Used for database connections and network connections
8. If static variable is modified it reflects globally.

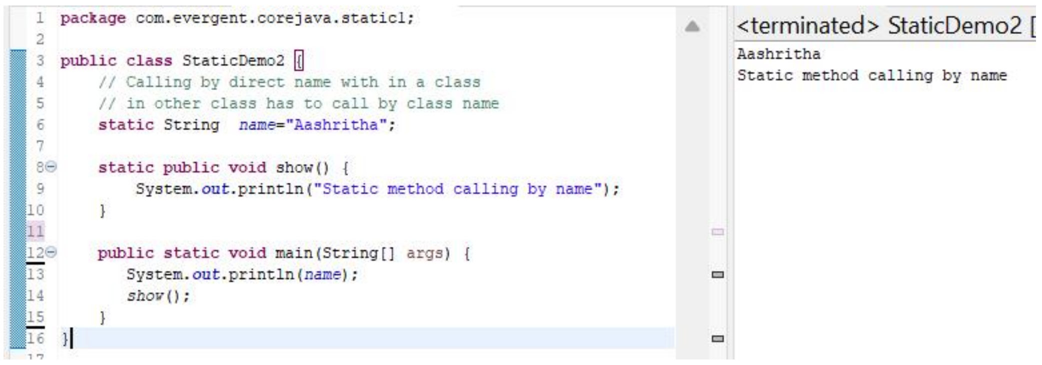
Points covered are: 1,2,3 [Calling by class name]



```java
package com.evergent.corejava.static1;

public class StaticDemo1 {
    //Calling by className
    static String name="Aashritha";

    static public void myData() {
        System.out.println("Static method");
    }
    public static void main(String[] args) {
        System.out.println(StaticDemo1.name);
        StaticDemo1.myData();
    }
}
```

```
<terminated> Sta
Aashritha
Static method
```

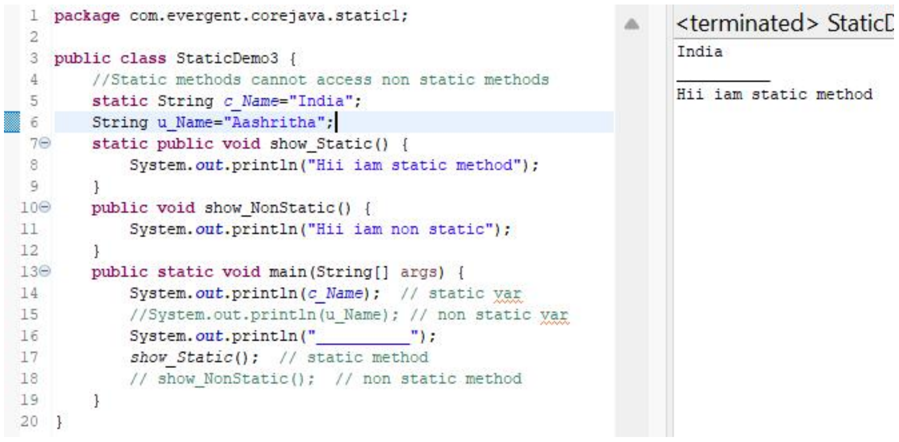
2. **Points covered are: 1,2,3 [Calling by direct name]**



```java
package com.evergent.corejava.static1;

public class StaticDemo2 {
    // Calling by direct name with in a class
    // in other class has to call by class name
    static String  name="Aashritha";

    static public void show() {
        System.out.println("Static method calling by name");
    }

    public static void main(String[] args) {
       System.out.println(name);
       show();
    }
}
```

```
<terminated> StaticDemo2 [
Aashritha
Static method calling by name
```

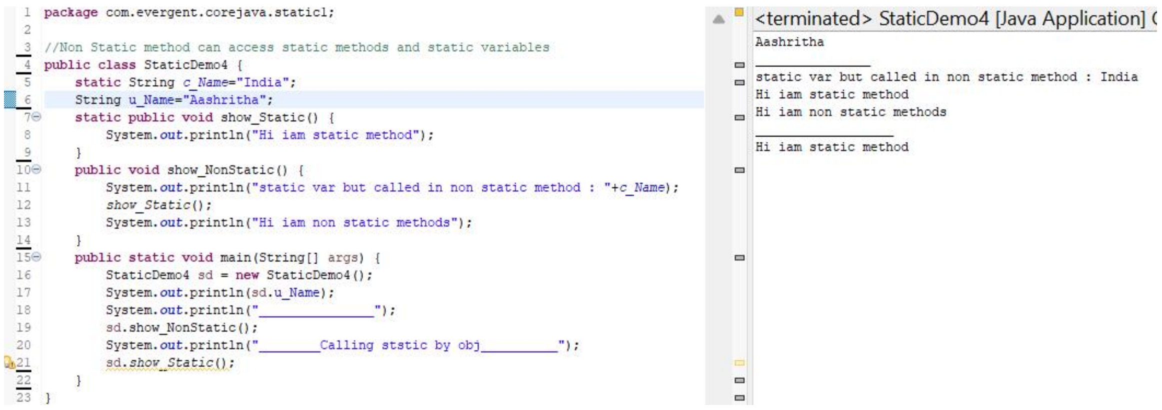
3. **Points covered are: 4,5**



```
package com.evergent.corejava.static1;

public class StaticDemo3 {
    //Static methods cannot access non static methods
    static String c_Name="India";
    String u_Name="Aashritha";
    static public void show_Static() {
        System.out.println("Hii iam static method");
    }
    public void show_NonStatic() {
        System.out.println("Hii iam non static");
    }
    public static void main(String[] args) {
        System.out.println(c_Name);  // static var
        //System.out.println(u_Name); // non static var
        System.out.println("__________");
        show_Static();  // static method
        // show_NonStatic();  // non static method
    }
}
```

```
<terminated> StaticD
India
__________
Hii iam static method
```

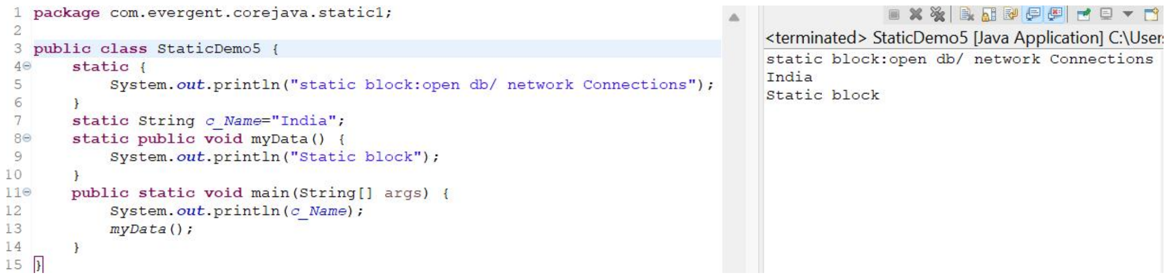
4. **Points covered are: 6**



```
package com.evergent.corejava.static1;

//Non Static method can access static methods and static variables
public class StaticDemo4 {
    static String c_Name="India";
    String u_Name="Aashritha";
    static public void show_Static() {
        System.out.println("Hi iam static method");
    }
    public void show_NonStatic() {
        System.out.println("static var but called in non static method : "+c_Name);
        show_Static();
        System.out.println("Hi iam non static methods");
    }
    public static void main(String[] args) {
        StaticDemo4 sd = new StaticDemo4();
        System.out.println(sd.u_Name);
        System.out.println("______________");
        sd.show_NonStatic();
        System.out.println("_______Calling ststic by obj_________");
        sd.show_Static();
    }
}
```

```
<terminated> StaticDemo4 [Java Application] (
Aashritha
______________
static var but called in non static method : India
Hi iam static method
Hi iam non static methods
______________
Hi iam static method
```

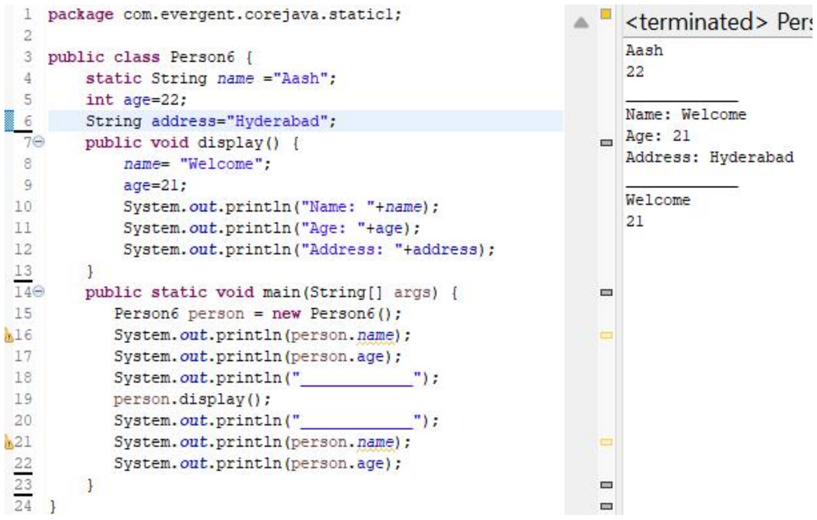
5. **Points covered are: 7**



```
package com.evergent.corejava.static1;

public class StaticDemo5 {
    static {
        System.out.println("static block:open db/ network Connections");
    }
    static String c_Name="India";
    static public void myData() {
        System.out.println("Static block");
    }
    public static void main(String[] args) {
        System.out.println(c_Name);
        myData();
    }
}
```

```
<terminated> StaticDemo5 [Java Application] C:\User
static block:open db/ network Connections
India
Static block
```

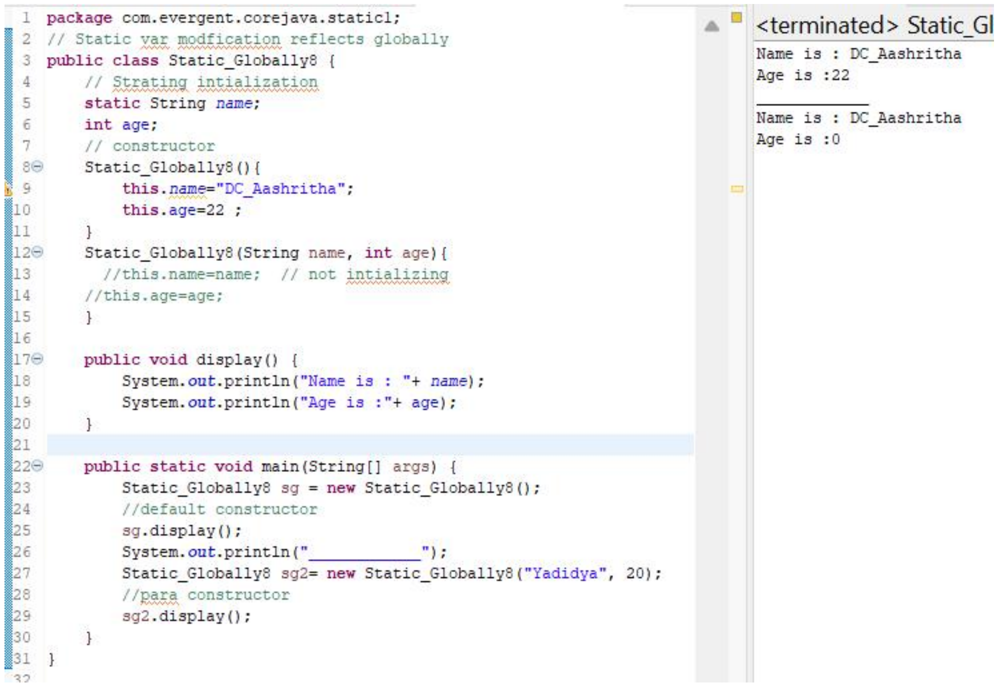
6. **Points covered are: 8**



```java
package com.evergent.corejava.static1;

public class Person6 {
    static String name ="Aash";
    int age=22;
    String address="Hyderabad";
    public void display() {
        name= "Welcome";
        age=21;
        System.out.println("Name: "+name);
        System.out.println("Age: "+age);
        System.out.println("Address: "+address);
    }
    public static void main(String[] args) {
       Person6 person = new Person6();
       System.out.println(person.name);
       System.out.println(person.age);
       System.out.println("____________");
       person.display();
       System.out.println("____________");
       System.out.println(person.name);
       System.out.println(person.age);
    }
}
```

```
<terminated> Pers
Aash
22
____________
Name: Welcome
Age: 21
Address: Hyderabad
____________
Welcome
21
```

7. **Points covered are: 8 [**Own example Static var modification reflects globlally**]**



```java
package com.evergent.corejava.static1;
// Static var modfication reflects globally
public class Static_Globally8 {
    // Strating intialization
    static String name;
    int age;
    // constructor
    Static_Globally8(){
        this.name="DC_Aashritha";
        this.age=22 ;
    }
    Static_Globally8(String name, int age){
      //this.name=name;  // not intializing
    //this.age=age;
    }

    public void display() {
        System.out.println("Name is : "+ name);
        System.out.println("Age is :"+ age);
    }

    public static void main(String[] args) {
        Static_Globally8 sg = new Static_Globally8();
        //default constructor
        sg.display();
        System.out.println("____________");
        Static_Globally8 sg2= new Static_Globally8("Yadidya", 20);
        //para constructor
        sg2.display();
    }
}
```

```
<terminated> Static_Gl
Name is : DC_Aashritha
Age is :22
____________
Name is : DC_Aashritha
Age is :0
```

Final :-

Final is a keyword

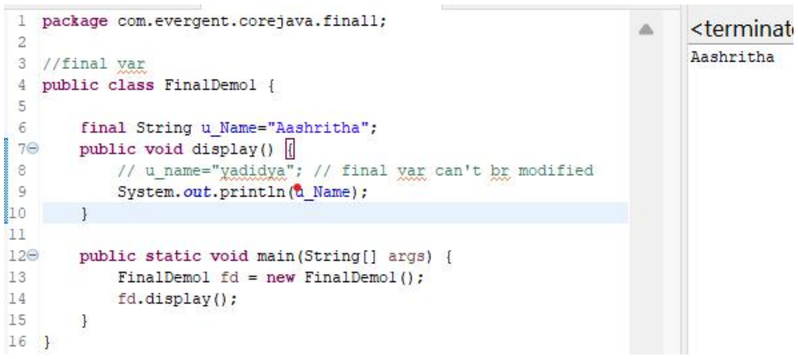
We can declare final as Instance & Local var, method and class

Final variables we can’t modify

Final methods cannot be overriden

Final class cannot be inherited by other classes but final class can inherit other classes.

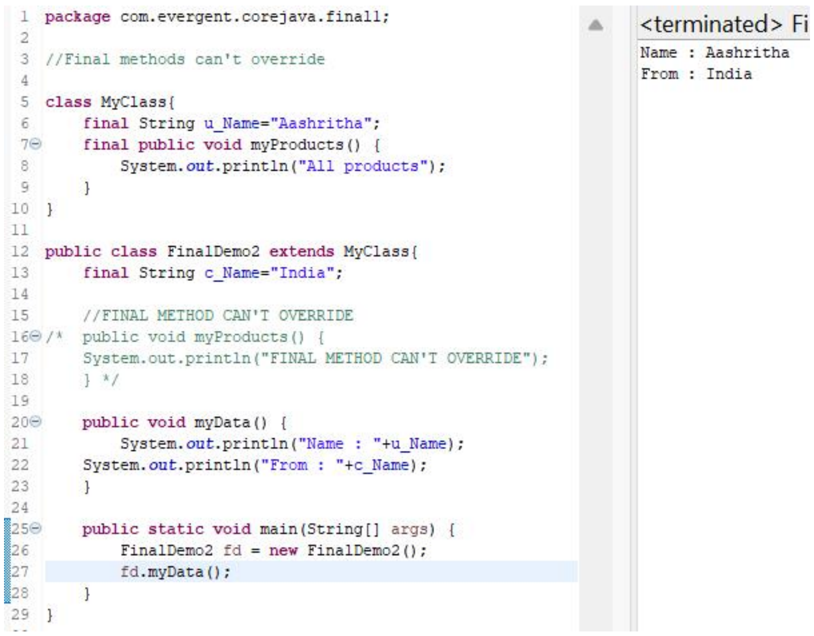
Points covered are: 1,2,3



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

//final var
public class FinalDemo1 {

    final String u_Name="Aashritha";
    public void display() {
        // u_name="vadidya"; // final var can't br modified
        System.out.println(u_Name);
    }

    public static void main(String[] args) {
        FinalDemo1 fd = new FinalDemo1();
        fd.display();
    }
}
```

```
<terminat
Aashritha
```

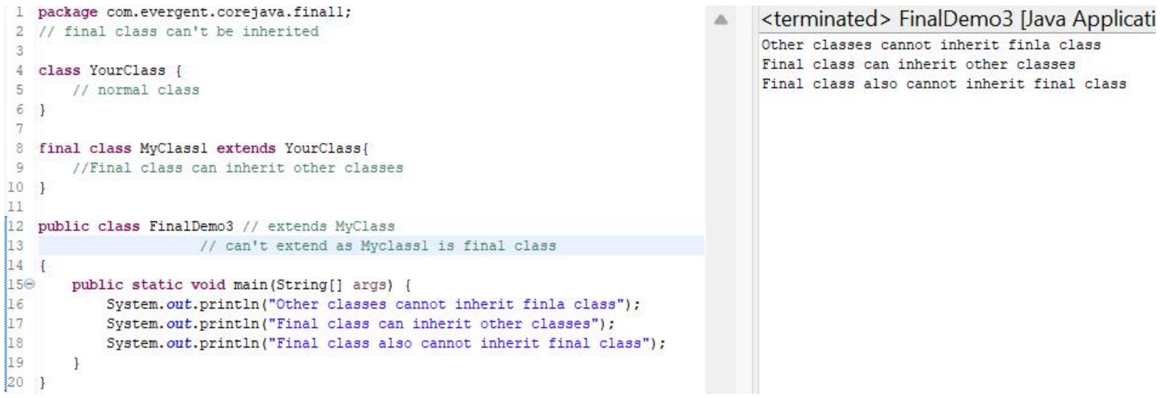
2. **Points covered are: 4**



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

//Final methods can't override

class MyClass{
    final String u_Name="Aashritha";
    final public void myProducts() {
        System.out.println("All products");
    }
}

public class FinalDemo2 extends MyClass{
    final String c_Name="India";

    //FINAL METHOD CAN'T OVERRIDE
/*  public void myProducts() {
    System.out.println("FINAL METHOD CAN'T OVERRIDE");
    } */

    public void myData() {
        System.out.println("Name : "+u_Name);
    System.out.println("From : "+c_Name);
    }

    public static void main(String[] args) {
        FinalDemo2 fd = new FinalDemo2();
        fd.myData();
    }
}
```

```
<terminated> Fi
Name : Aashritha
From : India
```

3. **Points covered are: 5**



```
package com.evergent.corejava.final1;
// final class can't be inherited

class YourClass {
    // normal class
}

final class MyClass1 extends YourClass{
    //Final class can inherit other classes
}

public class FinalDemo3 // extends MyClass
                   // can't extend as Myclass1 is final class
{
    public static void main(String[] args) {
        System.out.println("Other classes cannot inherit finla class");
        System.out.println("Final class can inherit other classes");
        System.out.println("Final class also cannot inherit final class");
    }
}
```

```
<terminated> FinalDemo3 [Java Applicati
Other classes cannot inherit finla class
Final class can inherit other classes
Final class also cannot inherit final class
```

DAY-6 (12/08/2024)

Strings :-

String class

String is a final class

String is immutable : Once we declare any string object it is constant if we are trying to modify existing string it will create other memory location, the exsisting object is eligible for garbage collection.

String class having methods

All string class methods are non synchronized

2) **String Buffer**

String Buffer is a final class

String Buffer is mutable

String Buffer having methods

append(), 2) insert(), 3) delete(), 4) replace(), 5) reverse(), 6) capacity(),

7) length()

iv. All String Buffer class methods are synchronized

v. **(= =)** --> To validate the address

vi. **(.equals())** -->It is a overriden method of object class(in obj class it will verify address)

In string class it will check the content

3) **String Builder**

String is a final class

String Builder is mutable

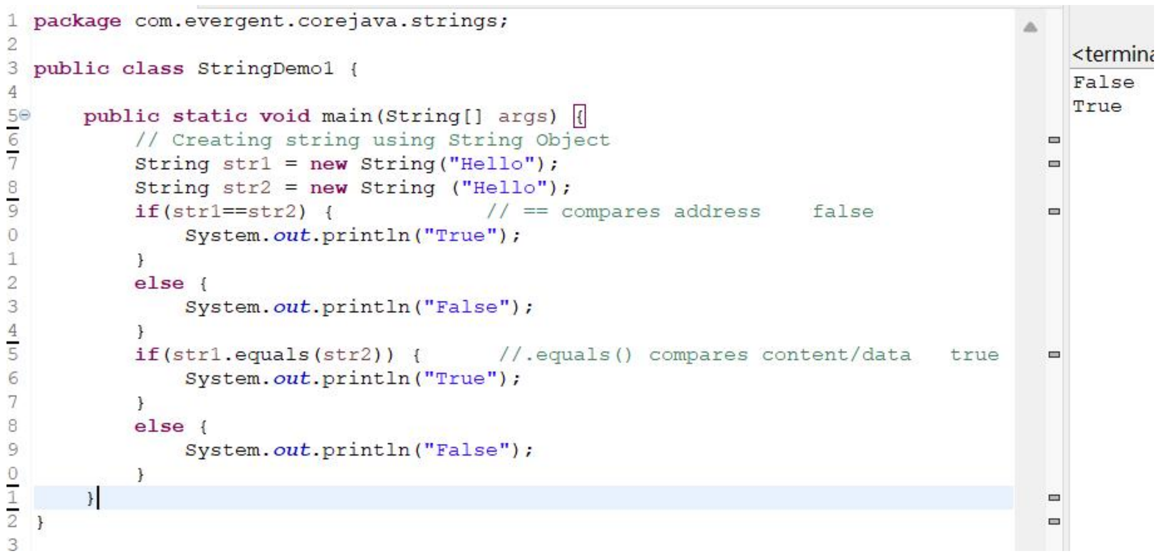
String Builder having methods

append(), 2) insert(), 3) delete(), 4) replace(), 5) reverse(), 6) capacity(),

7) length()

iv. All string class methods are non synchronized

Program 1:- String object == and .equals()

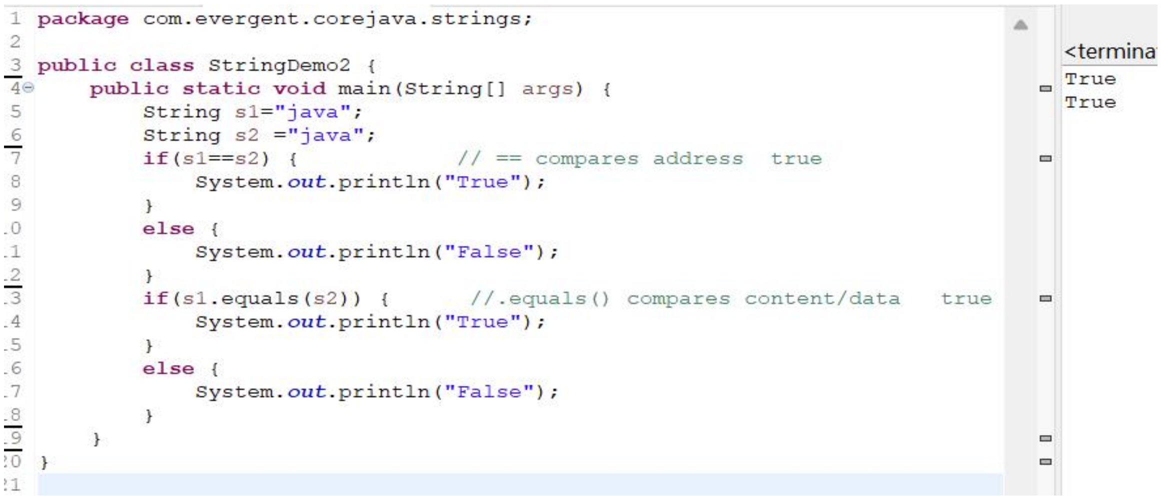


```
package com.evergent.corejava.strings;

public class StringDemo1 {

    public static void main(String[] args) {
        // Creating string using String Object
        String str1 = new String("Hello");
        String str2 = new String ("Hello");
        if(str1==str2) {            // == compares address    false
            System.out.println("True");
        }
        else {
            System.out.println("False");
        }
        if(str1.equals(str2)) {      //.equals() compares content/data   true
            System.out.println("True");
        }
        else {
            System.out.println("False");
        }
    }
}
```

<termina

```
False
True
```

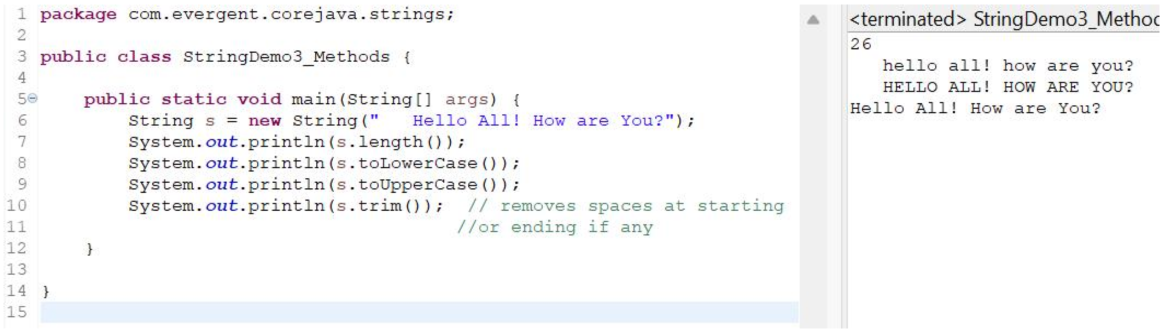
Program 2:- String literal == and .equals()



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

public class StringDemo2 {
    public static void main(String[] args) {
        String s1="java";
        String s2 ="java";
        if(s1==s2) {             // == compares address  true
            System.out.println("True");
        }
        else {
            System.out.println("False");
        }
        if(s1.equals(s2)) {       //.equals() compares content/data   true
            System.out.println("True");
        }
        else {
            System.out.println("False");
        }
    }
}
```

```
<termina
True
True
```

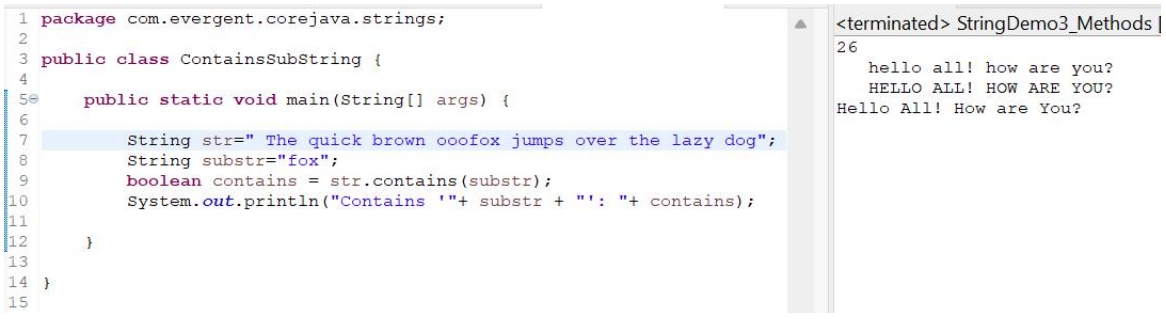
Program 3 :- String methods - length(), toLowercase(), toUppercase(), trim()



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

public class StringDemo3_Methods {

    public static void main(String[] args) {
        String s = new String("   Hello All! How are You?");
        System.out.println(s.length());
        System.out.println(s.toLowerCase());
        System.out.println(s.toUpperCase());
        System.out.println(s.trim());  // removes spaces at starting
                                       //or ending if any
    }

}
```

```
<terminated> StringDemo3_Methoc
26
   hello all! how are you?
   HELLO ALL! HOW ARE YOU?
Hello All! How are You?
```

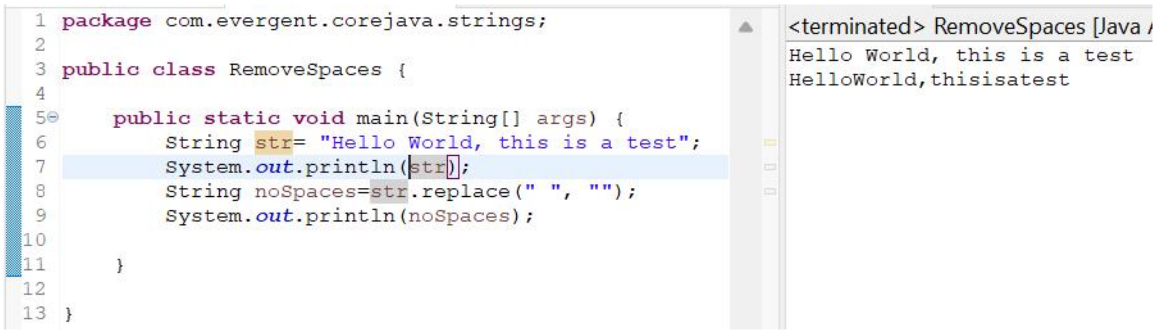
Program 4:- String methods - .contains()



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

public class ContainsSubString {

    public static void main(String[] args) {

        String str=" The quick brown ooofox jumps over the lazy dog";
        String substr="fox";
        boolean contains = str.contains(substr);
        System.out.println("Contains '"+ substr + "': "+ contains);

    }

}
```

```
<terminated> StringDemo3_Methods |
26
   hello all! how are you?
   HELLO ALL! HOW ARE YOU?
Hello All! How are You?
```

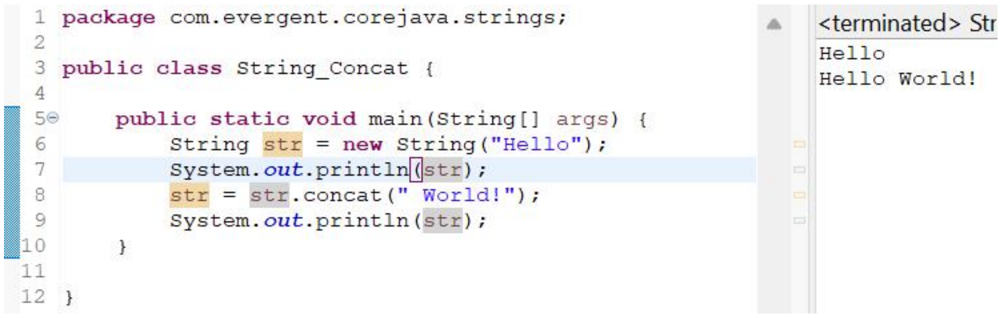
Program 4:- String methods - .replace()



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

public class RemoveSpaces {

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

    }

}
```

```
<terminated> RemoveSpaces [Java
Hello World, this is a test
HelloWorld,thisisatest
```

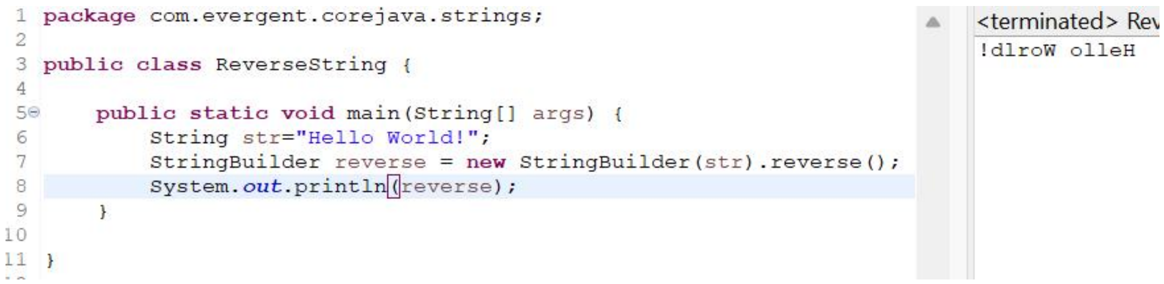
Program 5:- String methods - .concat()



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

public class String_Concat {

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

}
```

```
<terminated> Str
Hello
Hello World!
```

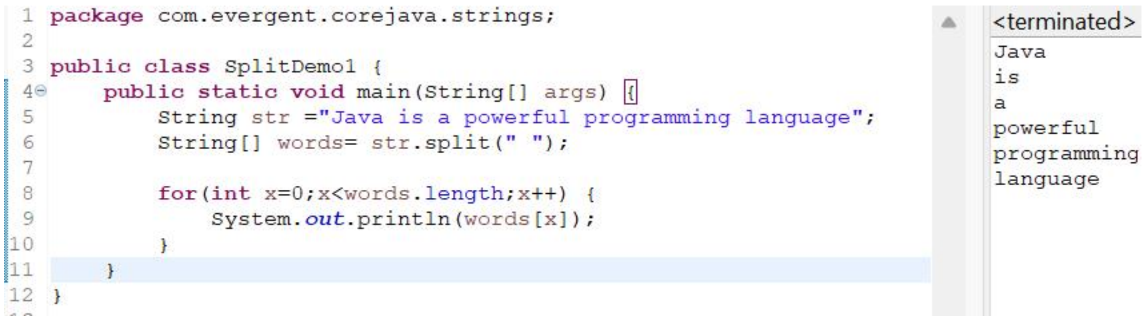
Program 6:- reversing string using STRING BUILDER method **. reverse()**



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

public class ReverseString {

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

}
```

```
<terminated> Rev
!dlroW olleH
```

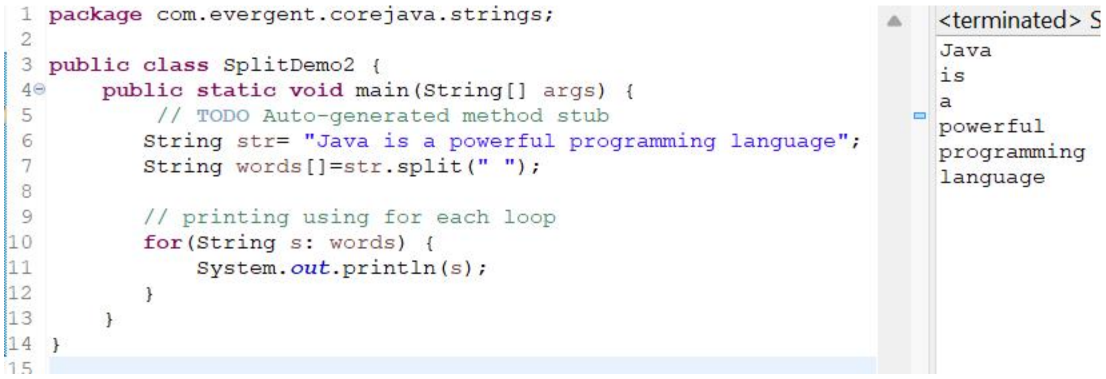
Program 7:- String methods - .split("parameter") using for loop



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

public class SplitDemo1 {
    public static void main(String[] args) {
        String str ="Java is a powerful programming language";
        String[] words= str.split(" ");

        for(int x=0;x<words.length;x++) {
            System.out.println(words[x]);
        }
    }
}
```

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

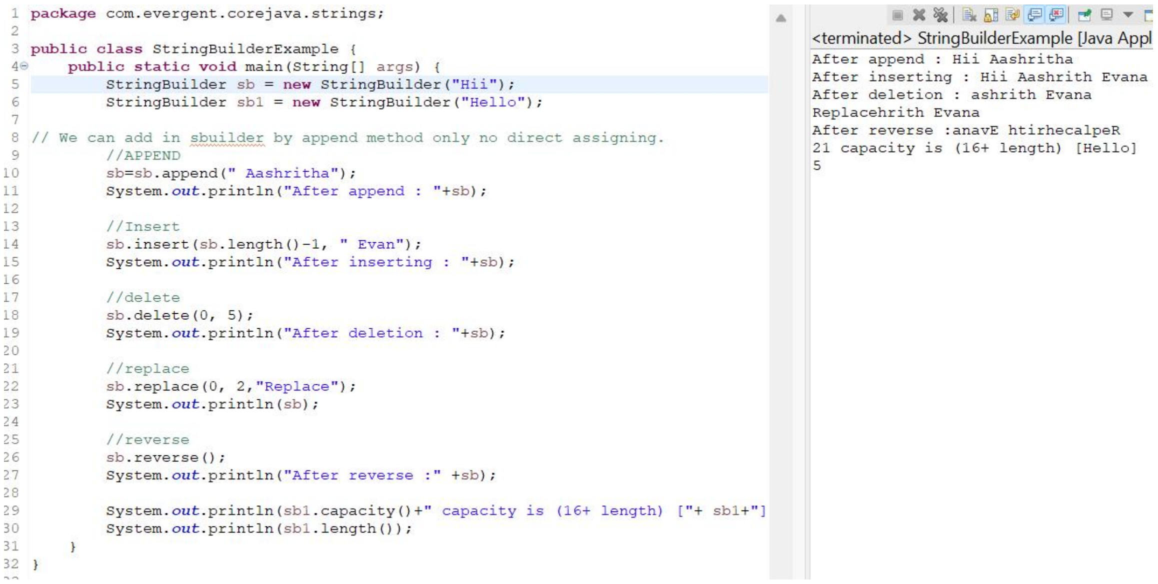
Program 8:- String methods - .split("parameter") using for each loop



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

public class SplitDemo2 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str= "Java is a powerful programming language";
        String words[]=str.split(" ");

        // printing using for each loop
        for(String s: words) {
            System.out.println(s);
        }
    }
}
```

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

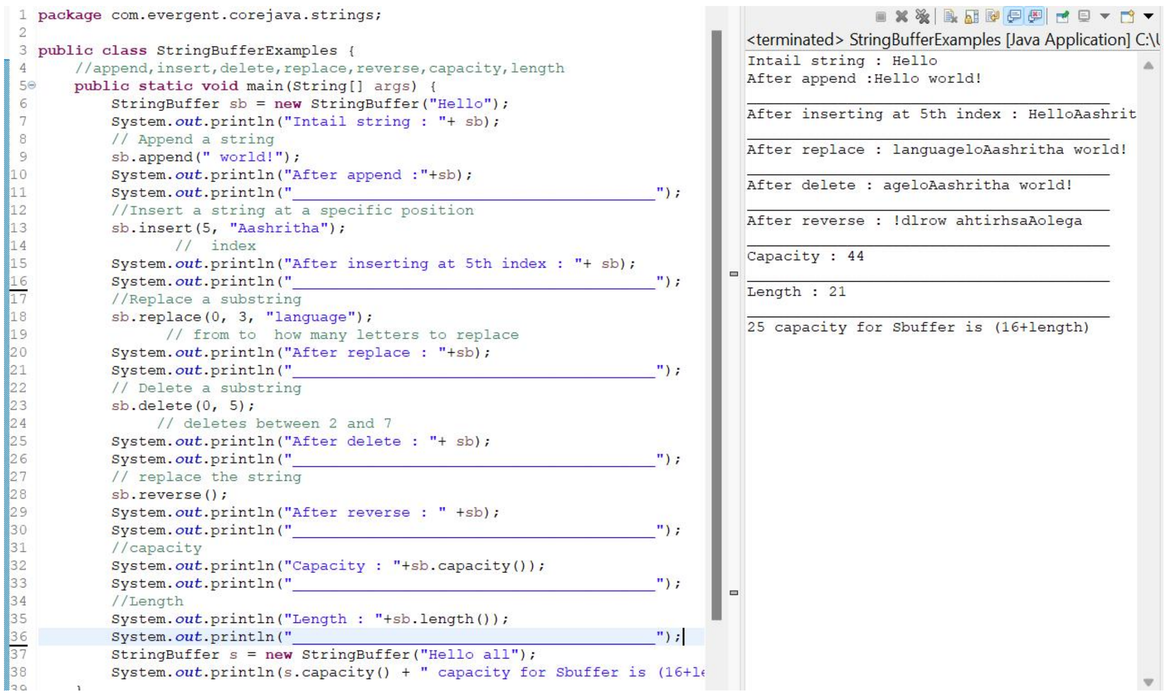
Program 9:- String Builder methods - append(), insert(), replace(), delete(), reverse(), capacity(), length()



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

public class StringBuilderExample {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder("Hii");
        StringBuilder sb1 = new StringBuilder("Hello");

// We can add in sbuilder by append method only no direct assigning.
        //APPEND
        sb=sb.append(" Aashritha");
        System.out.println("After append : "+sb);

        //Insert
        sb.insert(sb.length()-1, " Evan");
        System.out.println("After inserting : "+sb);

        //delete
        sb.delete(0, 5);
        System.out.println("After deletion : "+sb);

        //replace
        sb.replace(0, 2,"Replace");
        System.out.println(sb);

        //reverse
        sb.reverse();
        System.out.println("After reverse :" +sb);

        System.out.println(sb1.capacity()+" capacity is (16+ length) ["+ sb1+"]
        System.out.println(sb1.length());
    }
}
```

```
<terminated> StringBuilderExample [Java Appl
After append : Hii Aashritha
After inserting : Hii Aashrith Evana
After deletion : ashrith Evana
Replacehrith Evana
After reverse :anavE htirhecalpeR
21 capacity is (16+ length) [Hello]
5
```

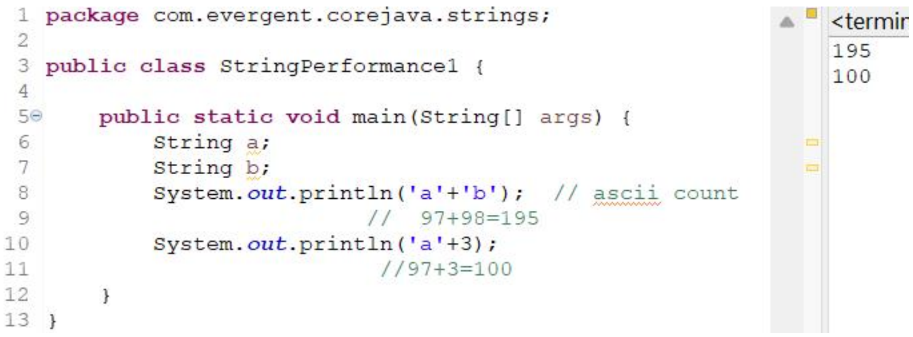
Program 10:- String Buffer methods - append(), insert(), replace(), delete(), reverse(), capacity(), length()



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

public class StringBufferExamples {
    //append,insert,delete,replace,reverse,capacity,length
    public static void main(String[] args) {
        StringBuffer sb = new StringBuffer("Hello");
        System.out.println("Intail string : "+ sb);
        // Append a string
        sb.append(" world!");
        System.out.println("After append :"+sb);
        System.out.println("________________________________________");
        //Insert a string at a specific position
        sb.insert(5, "Aashritha");
               //  index
        System.out.println("After inserting at 5th index : "+ sb);
        System.out.println("________________________________________");
        //Replace a substring
        sb.replace(0, 3, "language");
              // from to  how many letters to replace
        System.out.println("After replace : "+sb);
        System.out.println("________________________________________");
        // Delete a substring
        sb.delete(0, 5);
             // deletes between 2 and 7
        System.out.println("After delete : "+ sb);
        System.out.println("________________________________________");
        // replace the string
        sb.reverse();
        System.out.println("After reverse : " +sb);
        System.out.println("________________________________________");
        //capacity
        System.out.println("Capacity : "+sb.capacity());
        System.out.println("________________________________________");
        //Length
        System.out.println("Length : "+sb.length());
        System.out.println("________________________________________");
        StringBuffer s = new StringBuffer("Hello all");
        System.out.println(s.capacity() + " capacity for Sbuffer is (16+le
    }
```

```
<terminated> StringBufferExamples [Java Application] C:\
Intail string : Hello
After append :Hello world!
________________________________________
After inserting at 5th index : HelloAashrit
________________________________________
After replace : languageloAashritha world!
________________________________________
After delete : ageloAashritha world!
________________________________________
After reverse : !dlrow ahtirhsaAolega
________________________________________
Capacity : 44
________________________________________
Length : 21
________________________________________
25 capacity for Sbuffer is (16+length)
```

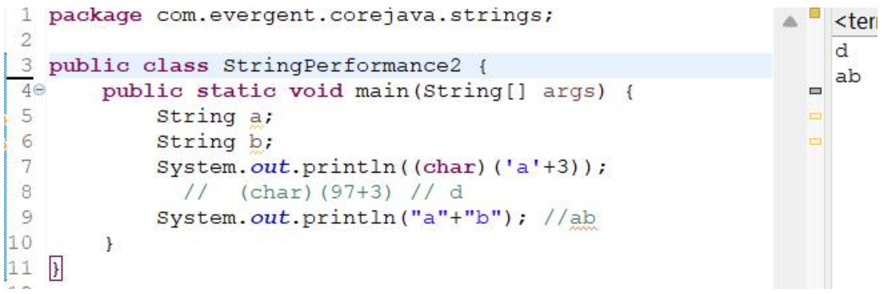
Program 11:- String Performance(Addition of char's**)**



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

public class StringPerformance1 {

    public static void main(String[] args) {
        String a;
        String b;
        System.out.println('a'+'b');  // ascii count
                         //  97+98=195
        System.out.println('a'+3);
                          //97+3=100
    }
}
```

```
<termir
195
100
```

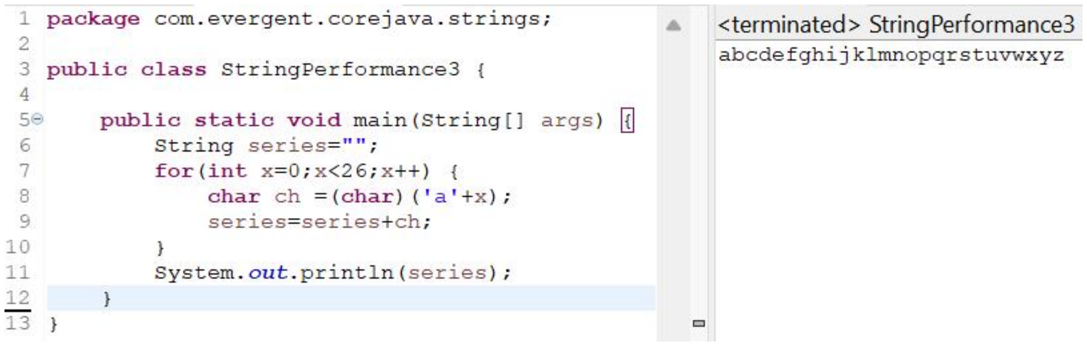
Program 12:- String Performance(Addition of char's & changing addition number into char**)**



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

public class StringPerformance2 {
    public static void main(String[] args) {
        String a;
        String b;
        System.out.println((char)('a'+3));
          //  (char)(97+3) // d
        System.out.println("a"+"b"); //ab
    }
}
```

```
<ter
d
ab
```

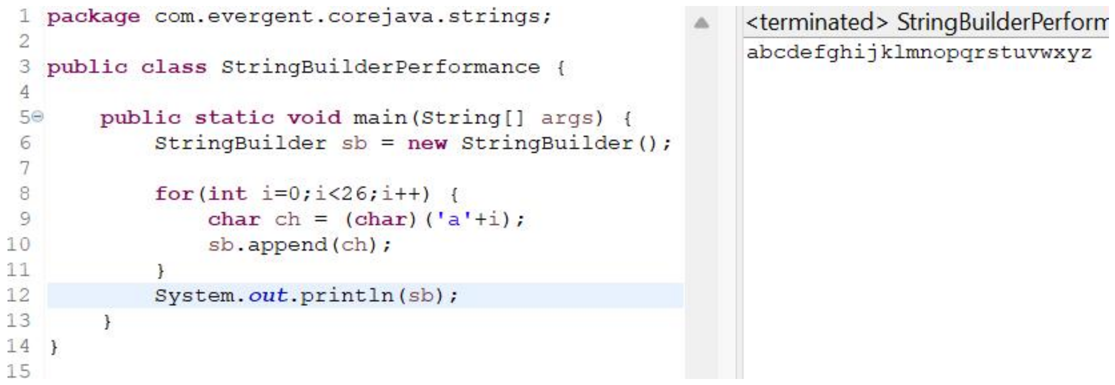
Program 13:- String Performance(series of alphabets **)**



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

public class StringPerformance3 {

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

```
<terminated> StringPerformance3
abcdefghijklmnopqrstuvwxyz
```

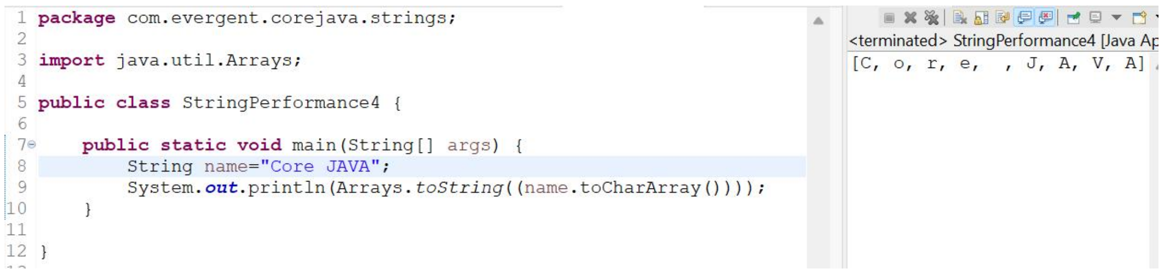
Program 14:- StringBuilder Performance(series of alphabets **)**



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

public class StringBuilderPerformance {

    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.println(sb);
    }
}

```

```
<terminated> StringBuilderPerform
abcdefghijklmnopqrstuvwxyz
```

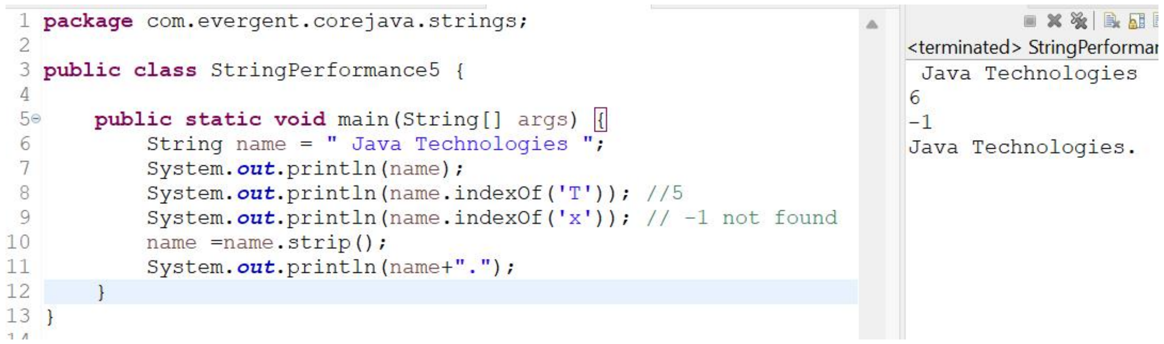
Program 15:- String Performance(toString() & toCharArray()**)**



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

import java.util.Arrays;

public class StringPerformance4 {

    public static void main(String[] args) {
        String name="Core JAVA";
        System.out.println(Arrays.toString((name.toCharArray())));
    }

}
```

```
<terminated> StringPerformance4 [Java Ap
[C, o, r, e,  , J, A, V, A]
```

Program 16:- String Performance(indexOf(), strip()**)**



```
package com.evergent.corejava.strings;

public class StringPerformance5 {

    public static void main(String[] args) {
        String name = " Java Technologies ";
        System.out.println(name);
        System.out.println(name.indexOf('T')); //5
        System.out.println(name.indexOf('x')); // -1 not found
        name =name.strip();
        System.out.println(name+".");
    }
}
```

```
<terminated> StringPerformar
 Java Technologies
6
-1
Java Technologies.
```

4) **Difference between String, StringBuffer, StringBuilder**

STRING	STRING BUFFER	STRING BUILDER
String is immutable	String Buffer is mutable	String Buffer is mutable
String Can be created using new keyword and string literal	String Buffer Can be created using new keyword	String Builder Can be created using new keyword
Non-Synchronized	Synchronized	Non-Synchronized
Not thread safe	Thread safe	Not thread safe
Memory created in heap for new word string and in string literal/constant pool for string literal	Memory created in heap	Memory created in heap
Fast performance	Slow performance	Fast performance
Mostly used in developing	Not used for developing as it is outdated	Used for developing but most preference is given to string

5) String class important points

In java, a string is a sequence of characters, often used to represent a text.

Strings are objects in java, they are instances of string class and they belong to java.lang package

Key features of strings in java:

Immutable: Once a string object is created it cannot be changed, Any modification to a string creates

DAY-7 (13/08/2024)

Can you make your class as Immutable? (YES)

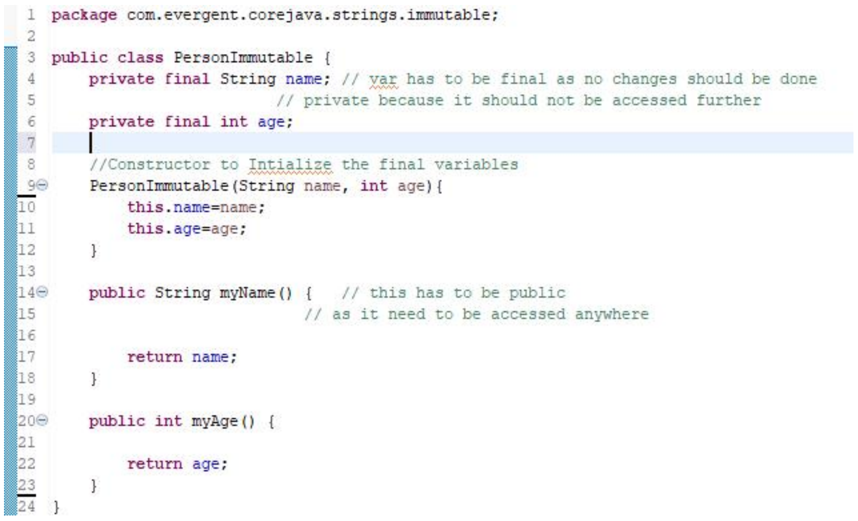
We can create our own immutable class by declaring

? class as final

? Methods as normal public

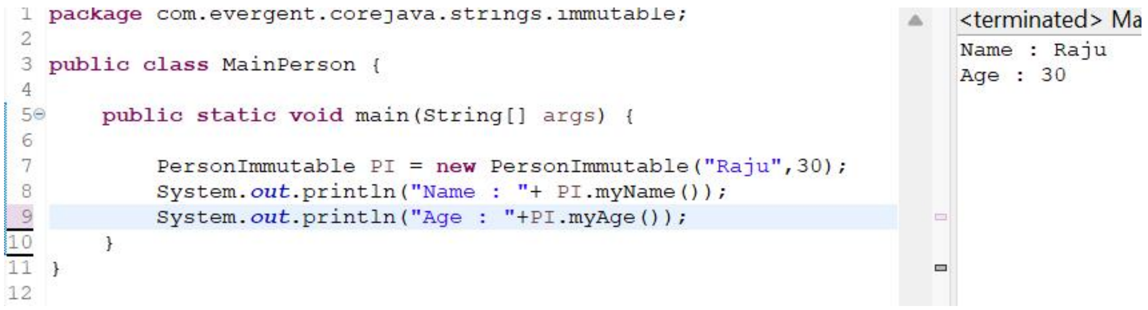
? Attributes/Variables as private and final

Immutable class --> step 1



```java
package com.evergent.corejava.strings.immutable;

public class PersonImmutable {
    private final String name; // var has to be final as no changes should be done
                                // private because it should not be accessed further
    private final int age;
    
    //Constructor to Intialize the final variables
    PersonImmutable(String name, int age){
        this.name=name;
        this.age=age;
    }

    public String myName() {    // this has to be public
                                // as it need to be accessed anywhere

        return name;
    }

    public int myAge() {

        return age;
    }
}
```

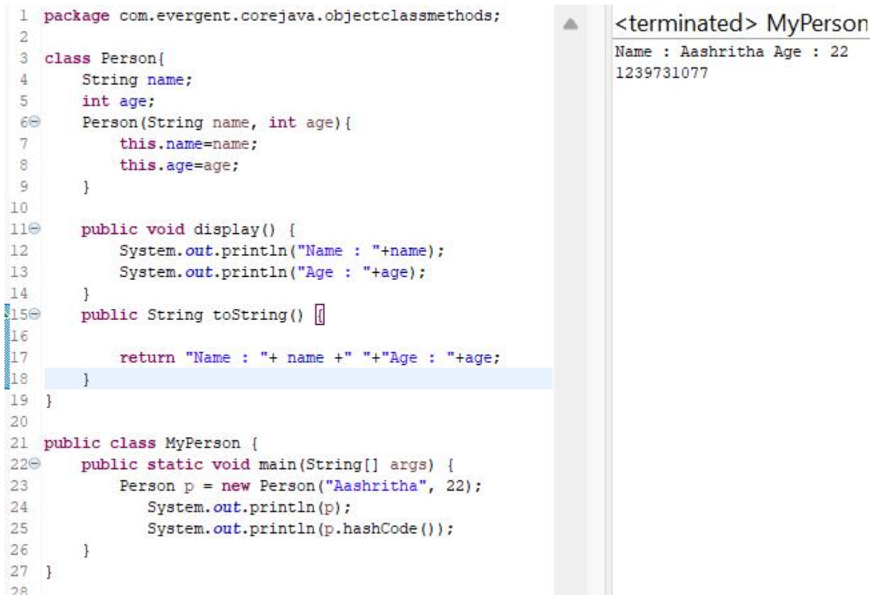
ii) Main class --> step-2



```java
package com.evergent.corejava.strings.immutable;

public class MainPerson {

    public static void main(String[] args) {

        PersonImmutable PI = new PersonImmutable("Raju",30);
        System.out.println("Name : "+ PI.myName());
        System.out.println("Age : "+PI.myAge());
    }
}

```

```
<terminated> Ma
Name : Raju
Age : 30
```

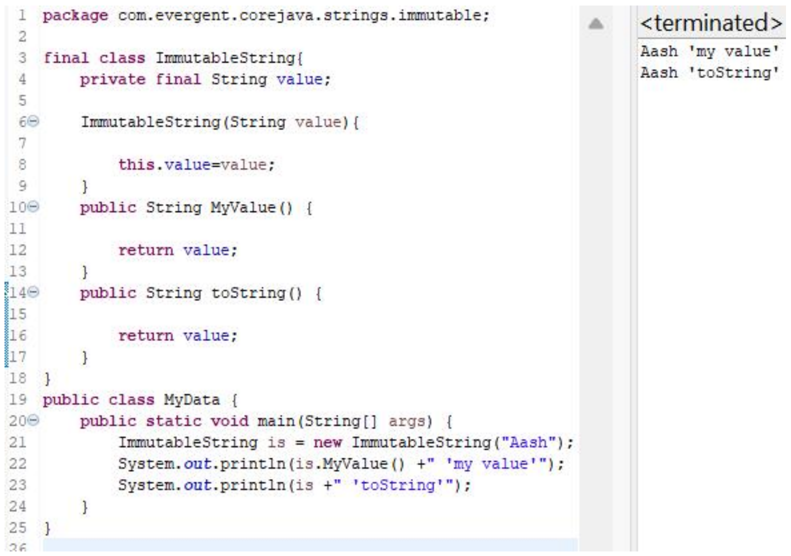
Object class methods [toString(), hashCode()]



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

class Person{
    String name;
    int age;
    Person(String name, int age){
        this.name=name;
        this.age=age;
    }

    public void display() {
        System.out.println("Name : "+name);
        System.out.println("Age : "+age);
    }
    public String toString() {

        return "Name : "+ name +" "+"Age : "+age;
    }
}

public class MyPerson {
    public static void main(String[] args) {
        Person p = new Person("Aashritha", 22);
           System.out.println(p);
           System.out.println(p.hashCode());
    }
}
```

```
<terminated> MyPerson
Name : Aashritha Age : 22
1239731077
```

3. **toString() method in StringImmutable class**



```java
package com.evergent.corejava.strings.immutable;

final class ImmutableString{
    private final String value;

    ImmutableString(String value){

        this.value=value;
    }
    public String MyValue() {

        return value;
    }
    public String toString() {

        return value;
    }
}
public class MyData {
    public static void main(String[] args) {
        ImmutableString is = new ImmutableString("Aash");
        System.out.println(is.MyValue() +" 'my value'");
        System.out.println(is +" 'toString'");
    }
}
```

```
<terminated>
Aash 'my value'
Aash 'toString'
```

DAY-10 (19/08/2024)

EXCEPTION HANDLING

Exception handling is mechanism

Exeptions are inbulit mechanism of java

All exceptions are executed while abnormal conditions only

Normal flow it won't execute any exceptions

Once any exceptions are occuring in java then remain lines of code is unreachable

Java.lang.Throwable is a super class for Exception and Errors

There are 2 types of exceptions in java

Checked Exception

Checked Exception

8. All checked exceptions are compile time exceptions
9. All unchecked exceptions are Runtime exceptions
10. There are 5 keywords in exception handling

try

catch()

finally()

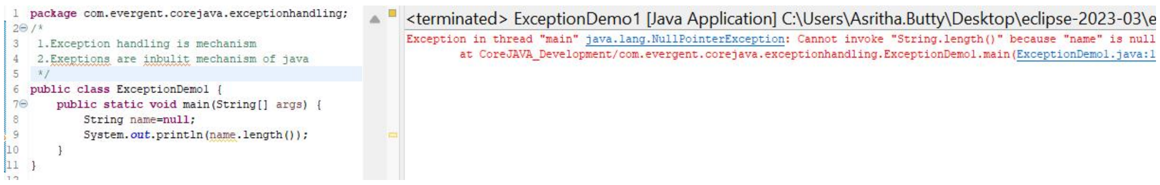
throws

throw

11. try is for business logic
12. catch is for handling exception
13. Finally is block, if exceptions is occurs or not finally block will be executed
14. throws an exception will be executed methods 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 Exception or Runtime exception
20. All exceptions classes are into java.lang package
21. There are two exceptions in class, Developer should be handle one after one
22. Developer can't handled errors
23. Handling multiple exceptions with throws

24. Handling multiple exceptions using multiple catch
25. In java, a nested try-catch block is try catch block within another try catch block

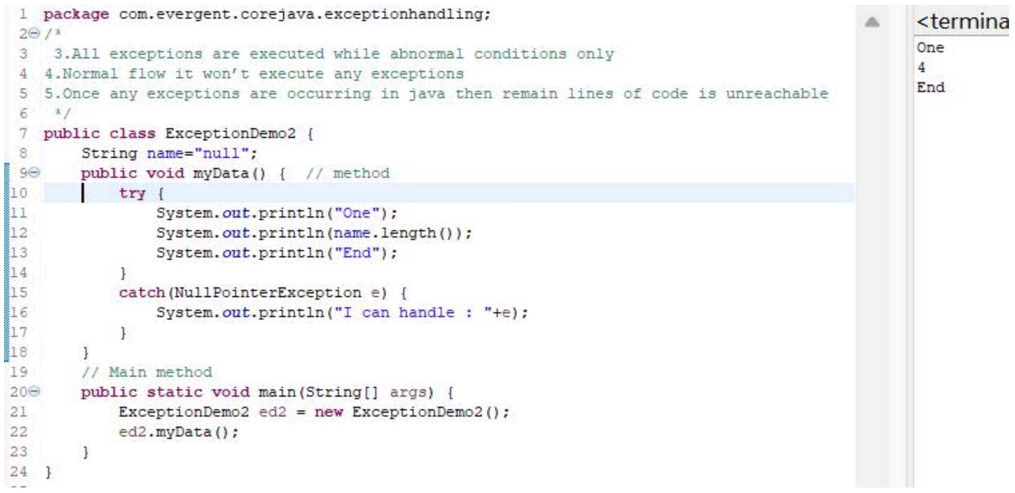
Points covered :- 1,2



```java
package com.evergent.corejava.exceptionhandling;
/*
 1.Exception handling is mechanism
 2.Exeptions are inbulit mechanism of java
 */
public class ExceptionDemo1 {
    public static void main(String[] args) {
        String name=null;
        System.out.println(name.length());
    }
}
```

```
<terminated> ExceptionDemo1 [Java Application] C:\Users\Asritha.Butty\Desktop\eclipse-2023-03\e
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" because "name" is null
	at CoreJAVA_Development/com.evergent.corejava.exceptionhandling.ExceptionDemo1.main(ExceptionDemo1.java:1
```

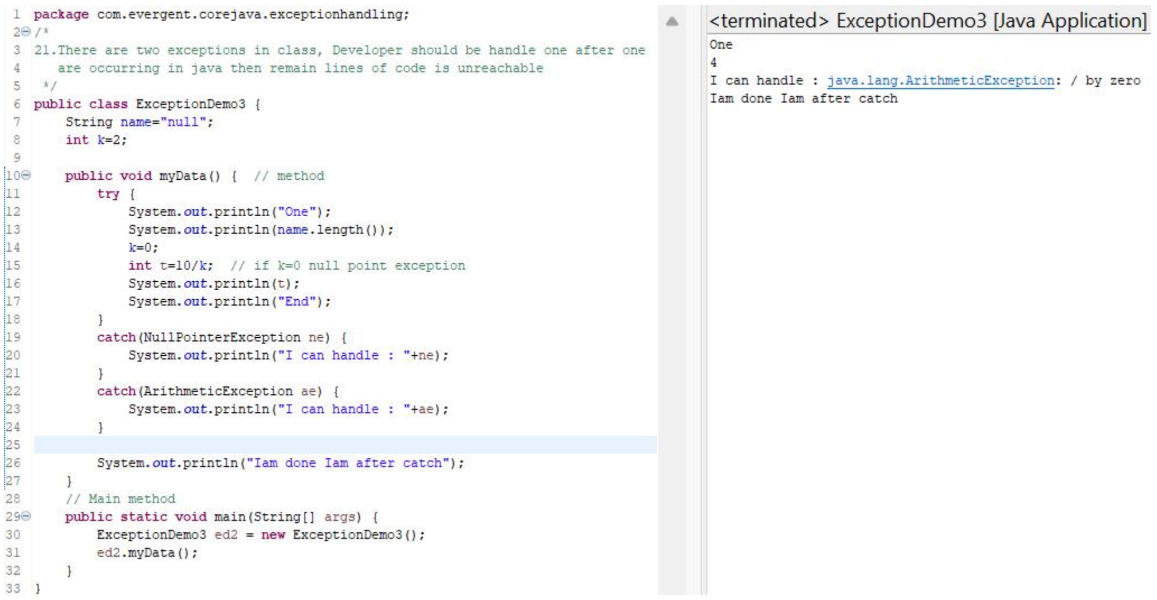
2. **Points Covered :- 3,4,5**



```java
package com.evergent.corejava.exceptionhandling;
/*
  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 remain lines of code is unreachable
  */
public class ExceptionDemo2 {
    String name="null";
    public void myData() {  // method
        try {
            System.out.println("One");
            System.out.println(name.length());
            System.out.println("End");
        }
        catch(NullPointerException e) {
            System.out.println("I can handle : "+e);
        }
    }
    // Main method
    public static void main(String[] args) {
        ExceptionDemo2 ed2 = new ExceptionDemo2();
        ed2.myData();
    }
}
```

```
<termina
One
4
End
```

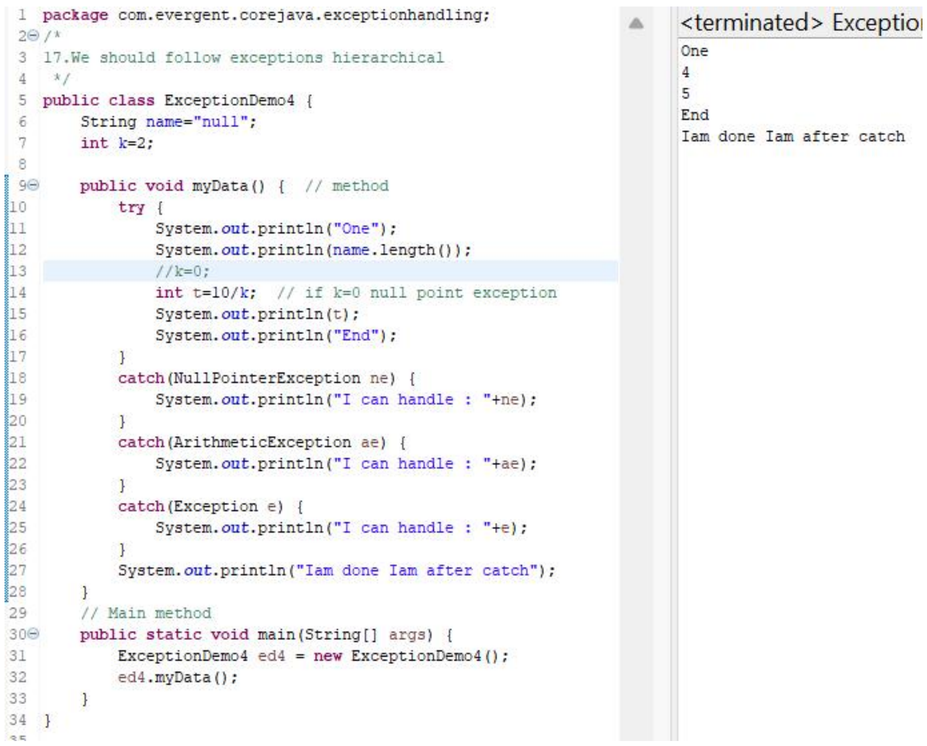
3. **Points Covered :- 21**



```java
package com.evergent.corejava.exceptionhandling;
/*
21.There are two exceptions in class, Developer should be handle one after one
   are occurring in java then remain lines of code is unreachable
 */
public class ExceptionDemo3 {
    String name="null";
    int k=2;

    public void myData() {  // method
        try {
            System.out.println("One");
            System.out.println(name.length());
            k=0;
            int t=10/k;  // if k=0 null point exception
            System.out.println(t);
            System.out.println("End");
        }
        catch(NullPointerException ne) {
            System.out.println("I can handle : "+ne);
        }
        catch(ArithmeticException ae) {
            System.out.println("I can handle : "+ae);
        }

        System.out.println("Iam done Iam after catch");
    }
    // Main method
    public static void main(String[] args) {
        ExceptionDemo3 ed2 = new ExceptionDemo3();
        ed2.myData();
    }
}
```

```
<terminated> ExceptionDemo3 [Java Application]
One
4
I can handle : java.lang.ArithmeticException: / by zero
Iam done Iam after catch
```

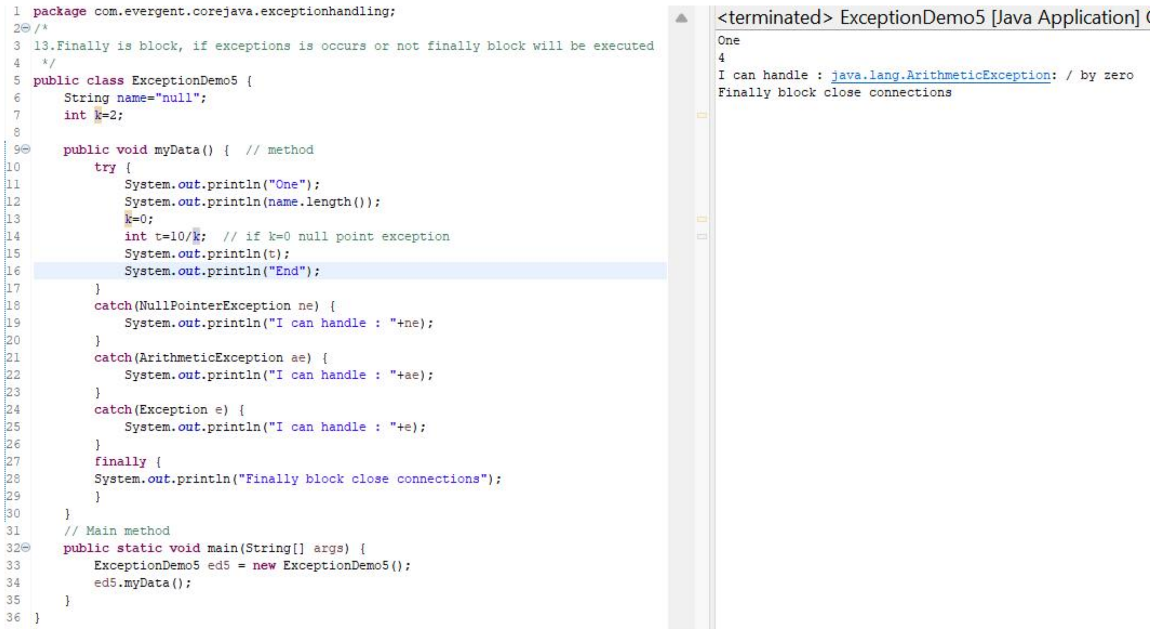
4. **Points Covered :- 17**



```java
package com.evergent.corejava.exceptionhandling;
/*
17.We should follow exceptions hierarchical
 */
public class ExceptionDemo4 {
    String name="null";
    int k=2;

    public void myData() {  // method
        try {
            System.out.println("One");
            System.out.println(name.length());
            //k=0;
            int t=10/k;  // if k=0 null point exception
            System.out.println(t);
            System.out.println("End");
        }
        catch(NullPointerException ne) {
            System.out.println("I can handle : "+ne);
        }
        catch(ArithmeticException ae) {
            System.out.println("I can handle : "+ae);
        }
        catch(Exception e) {
            System.out.println("I can handle : "+e);
        }
        System.out.println("Iam done Iam after catch");
    }
    // Main method
    public static void main(String[] args) {
        ExceptionDemo4 ed4 = new ExceptionDemo4();
        ed4.myData();
    }
}
```

```
<terminated> Exceptio
One
4
5
End
Iam done Iam after catch
```

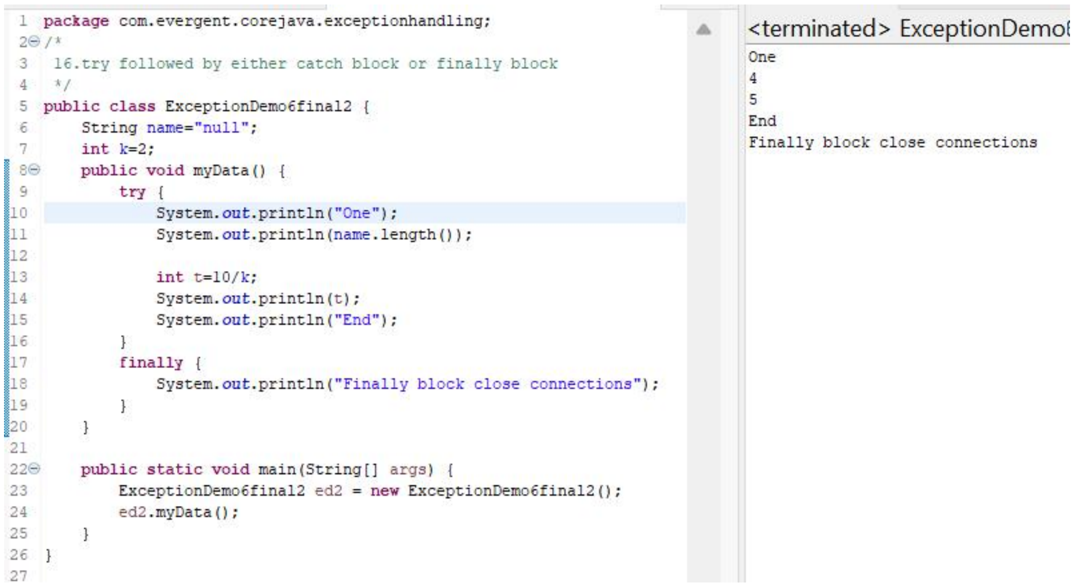
5. **Points Covered :- 17**



```java
package com.evergent.corejava.exceptionhandling;
/*
13.Finally is block, if exceptions is occurs or not finally block will be executed
 */
public class ExceptionDemo5 {
    String name="null";
    int k=2;

    public void myData() {  // method
        try {
            System.out.println("One");
            System.out.println(name.length());
            k=0;
            int t=10/k;  // if k=0 null point exception
            System.out.println(t);
            System.out.println("End");
        }
        catch(NullPointerException ne) {
            System.out.println("I can handle : "+ne);
        }
        catch(ArithmeticException ae) {
            System.out.println("I can handle : "+ae);
        }
        catch(Exception e) {
            System.out.println("I can handle : "+e);
        }
        finally {
        System.out.println("Finally block close connections");
        }
    }
    // Main method
    public static void main(String[] args) {
        ExceptionDemo5 ed5 = new ExceptionDemo5();
        ed5.myData();
    }
}
```

```
<terminated> ExceptionDemo5 [Java Application] (
One
4
I can handle : java.lang.ArithmeticException: / by zero
Finally block close connections
```

6. **Points Covered :- 16**

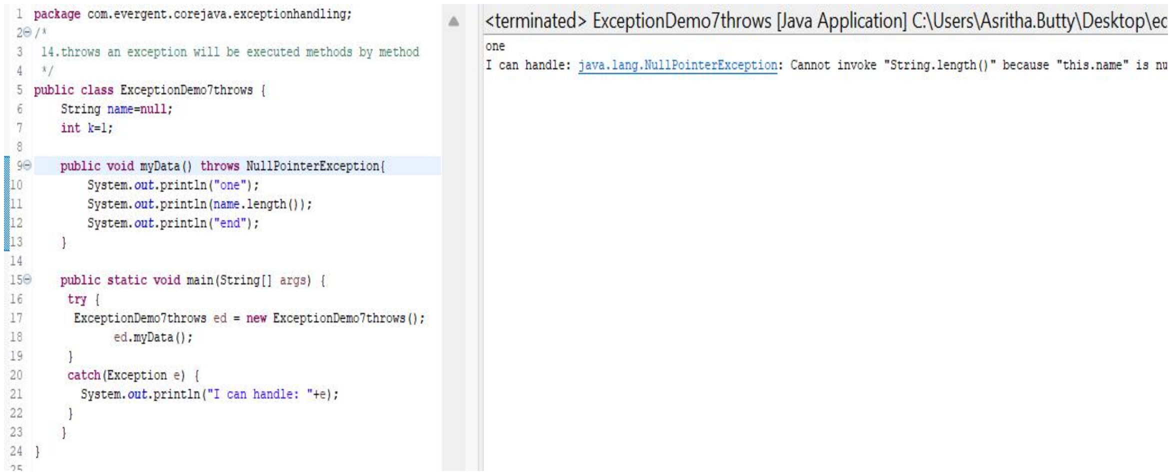


```
package com.evergent.corejava.exceptionhandling;
/*
 16.try followed by either catch block or finally block
 */
public class ExceptionDemo6final2 {
    String name="null";
    int k=2;
    public void myData() {
        try {
            System.out.println("One");
            System.out.println(name.length());

            int t=10/k;
            System.out.println(t);
            System.out.println("End");
        }
        finally {
            System.out.println("Finally block close connections");
        }
    }

    public static void main(String[] args) {
        ExceptionDemo6final2 ed2 = new ExceptionDemo6final2();
        ed2.myData();
    }
}
```

```
<terminated> ExceptionDemo6
One
4
5
End
Finally block close connections
```

DAY-11 (20/08/2024)

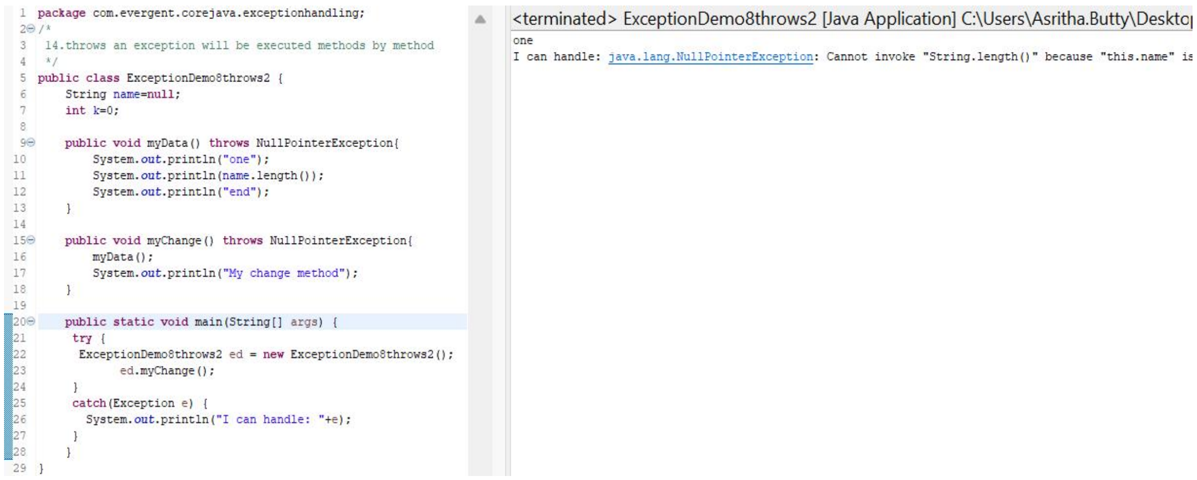
7. **Points Covered :- 14**



```
package com.evergent.corejava.exceptionhandling;
/*
 14.throws an exception will be executed methods by method
 */
public class ExceptionDemo7throws {
    String name=null;
    int k=1;

    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);
     }
    }
}
```

```
<terminated> ExceptionDemo7throws [Java Application] C:\Users\Asritha.Butty\Desktop\ec
one
I can handle: java.lang.NullPointerException: Cannot invoke "String.length()" because "this.name" is nu
```

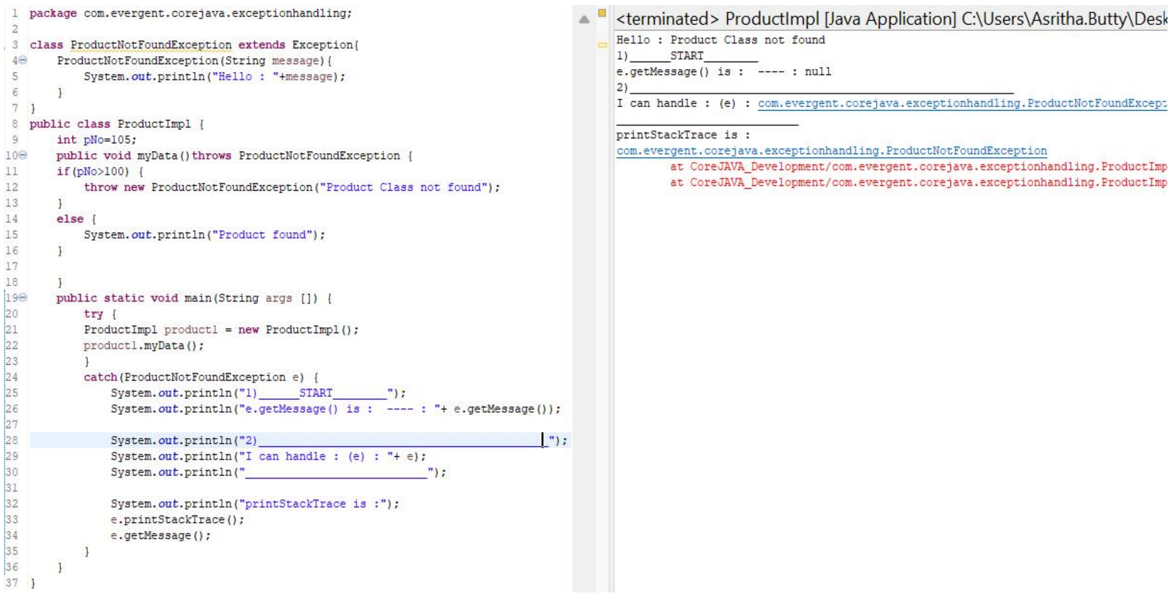
8. **Points Covered :- 14**



```java
package com.evergent.corejava.exceptionhandling;
/*
 14.throws an exception will be executed methods by method
 */
public class ExceptionDemo8throws2 {
    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 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> ExceptionDemo8throws2 [Java Application] C:\Users\Asritha.Butty\Desktop
one
I can handle: java.lang.NullPointerException: Cannot invoke "String.length()" because "this.name" is
```

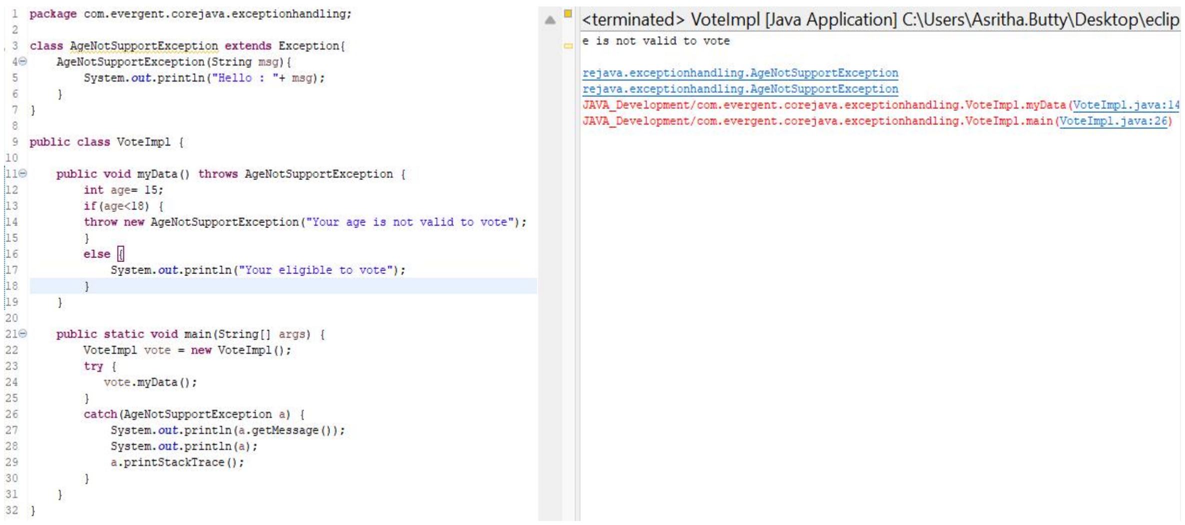
9. **Points Covered :- User Defined exceptions(ClassNotFoundException)**



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

class ProductNotFoundException extends Exception{
    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("Product Class not found");
    }
    else {
        System.out.println("Product found");
    }

    }
    public static void main(String args []) {
        try {
        ProductImpl product1 = new ProductImpl();
        product1.myData();
        }
        catch(ProductNotFoundException e) {
            System.out.println("1)______START________");
            System.out.println("e.getMessage() is :  ---- : "+ e.getMessage());

            System.out.println("2)_____________________________________________");
            System.out.println("I can handle : (e) : "+ e);
            System.out.println("___________________________");

            System.out.println("printStackTrace is :");
            e.printStackTrace();
            e.getMessage();
        }
    }
}
```

```
<terminated> ProductImpl [Java Application] C:\Users\Asritha.Butty\Desk
Hello : Product Class not found
1)______START________
e.getMessage() is :  ---- : null
2)_____________________________________________
I can handle : (e) : com.evergent.corejava.exceptionhandling.ProductNotFoundExcept
___________________________
printStackTrace is :
com.evergent.corejava.exceptionhandling.ProductNotFoundException
        at CoreJAVA_Development/com.evergent.corejava.exceptionhandling.ProductImp
        at CoreJAVA_Development/com.evergent.corejava.exceptionhandling.ProductImp
```

10.1 Points Covered :- User Defined exceptions (AgeNotSupportException)

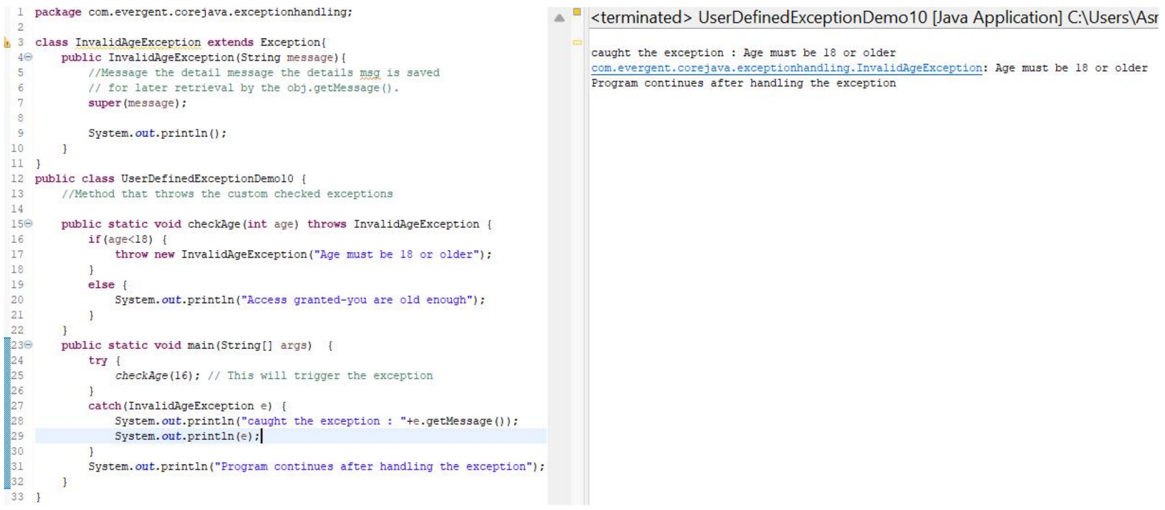


```java
package com.evergent.corejava.exceptionhandling;

class AgeNotSupportException extends Exception{
    AgeNotSupportException(String msg){
        System.out.println("Hello : "+ msg);
    }
}

public class VoteImpl {

    public void myData() throws AgeNotSupportException {
        int age= 15;
        if(age<18) {
        throw new AgeNotSupportException("Your age is not valid to vote");
        }
        else {
            System.out.println("Your eligible to vote");
        }
    }

    public static void main(String[] args) {
        VoteImpl vote = new VoteImpl();
        try {
           vote.myData();
        }
        catch(AgeNotSupportException a) {
            System.out.println(a.getMessage());
            System.out.println(a);
            a.printStackTrace();
        }
    }
}
```

```
<terminated> VoteImpl [Java Application] C:\Users\Asritha.Butty\Desktop\eclip
e is not valid to vote

rejava.exceptionhandling.AgeNotSupportException
rejava.exceptionhandling.AgeNotSupportException
JAVA_Development/com.evergent.corejava.exceptionhandling.VoteImpl.myData(VoteImpl.java:14
JAVA_Development/com.evergent.corejava.exceptionhandling.VoteImpl.main(VoteImpl.java:26)
```

DAY-12 (21/08/2024)

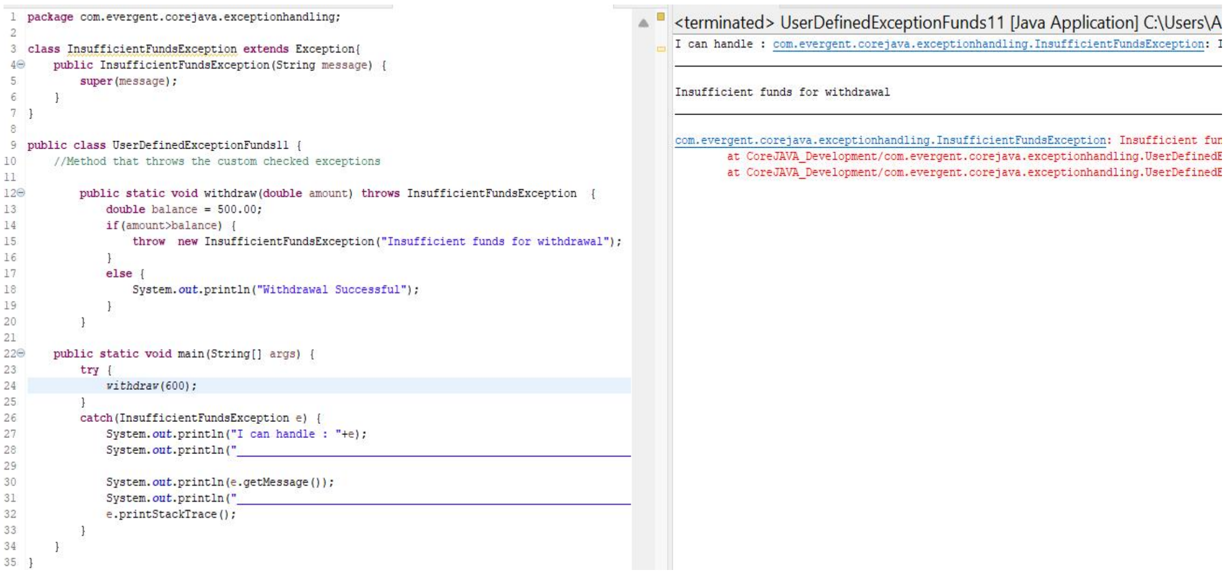
10.2. Points Covered :- User Defined exceptions (InvalidAgeException)



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

class InvalidAgeException extends Exception{
    public InvalidAgeException(String message){
        //Message the detail message the details msg is saved
        // for later retrieval by the obj.getMessage().
        super(message);

        System.out.println();
    }
}
public class UserDefinedExceptionDemo10 {
    //Method that throws the custom checked exceptions

    public static void checkAge(int age) throws InvalidAgeException {
        if(age<18) {
            throw new InvalidAgeException("Age must be 18 or older");
        }
        else {
            System.out.println("Access granted-you are old enough");
        }
    }
    public static void main(String[] args)  {
        try {
            checkAge(16); // This will trigger the exception
        }
        catch(InvalidAgeException e) {
            System.out.println("caught the exception : "+e.getMessage());
            System.out.println(e);
        }
        System.out.println("Program continues after handling the exception");
    }
}
```

```
<terminated> UserDefinedExceptionDemo10 [Java Application] C:\Users\Asr

caught the exception : Age must be 18 or older
com.evergent.corejava.exceptionhandling.InvalidAgeException: Age must be 18 or older
Program continues after handling the exception
```

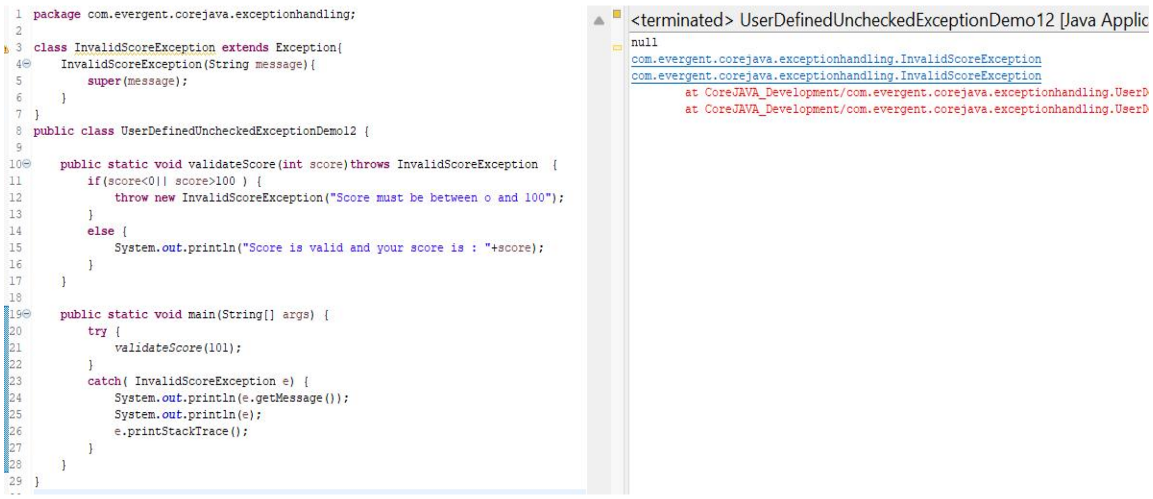
11. Points Covered :- User Defined exceptions (InsufficientFundsException)



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

class InsufficientFundsException extends Exception{
    public InsufficientFundsException(String message) {
        super(message);
    }
}

public class UserDefinedExceptionFunds11 {
    //Method that throws the custom checked exceptions

        public static void withdraw(double amount) throws InsufficientFundsException  {
            double balance = 500.00;
            if(amount>balance) {
                throw  new InsufficientFundsException("Insufficient funds for withdrawal");
            }
            else {
                System.out.println("Withdrawal Successful");
            }
        }

    public static void main(String[] args) {
        try {
            withdraw(600);
        }
        catch(InsufficientFundsException e) {
            System.out.println("I can handle : "+e);
            System.out.println("____________________________________________________________

            System.out.println(e.getMessage());
            System.out.println("____________________________________________________________
            e.printStackTrace();
        }
    }
}
```

```
<terminated> UserDefinedExceptionFunds11 [Java Application] C:\Users\A
I can handle : com.evergent.corejava.exceptionhandling.InsufficientFundsException: I
________________________________________________________________________________

Insufficient funds for withdrawal
________________________________________________________________________________

com.evergent.corejava.exceptionhandling.InsufficientFundsException: Insufficient fun
        at CoreJAVA_Development/com.evergent.corejava.exceptionhandling.UserDefinedE
        at CoreJAVA_Development/com.evergent.corejava.exceptionhandling.UserDefinedE
```

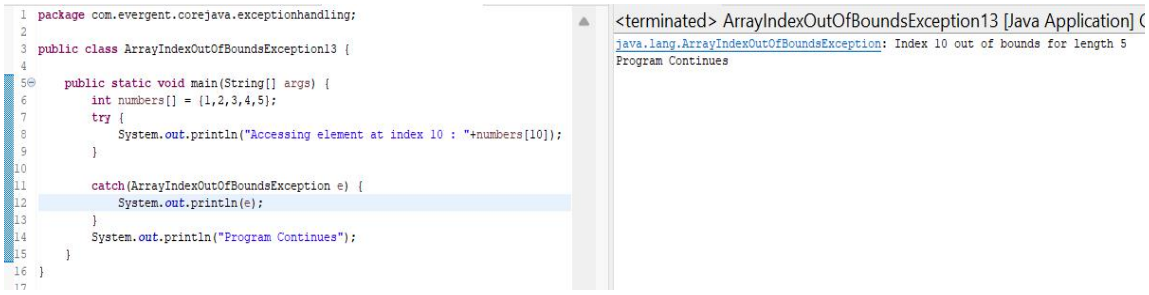
12. Points Covered :- User Defined exceptions (InvalidScoreException)



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

class InvalidScoreException extends Exception{
    InvalidScoreException(String message){
        super(message);
    }
}
public class UserDefinedUncheckedExceptionDemo12 {

    public static void validateScore(int score)throws InvalidScoreException  {
        if(score<0|| score>100 ) {
            throw new InvalidScoreException("Score must be between o and 100");
        }
        else {
            System.out.println("Score is valid and your score is : "+score);
        }
    }

    public static void main(String[] args) {
        try {
            validateScore(101);
        }
        catch( InvalidScoreException e) {
            System.out.println(e.getMessage());
            System.out.println(e);
            e.printStackTrace();
        }
    }
}
```

```
<terminated> UserDefinedUncheckedExceptionDemo12 [Java Applic
null
com.evergent.corejava.exceptionhandling.InvalidScoreException
com.evergent.corejava.exceptionhandling.InvalidScoreException
        at CoreJAVA_Development/com.evergent.corejava.exceptionhandling.UserD
        at CoreJAVA_Development/com.evergent.corejava.exceptionhandling.UserD
```

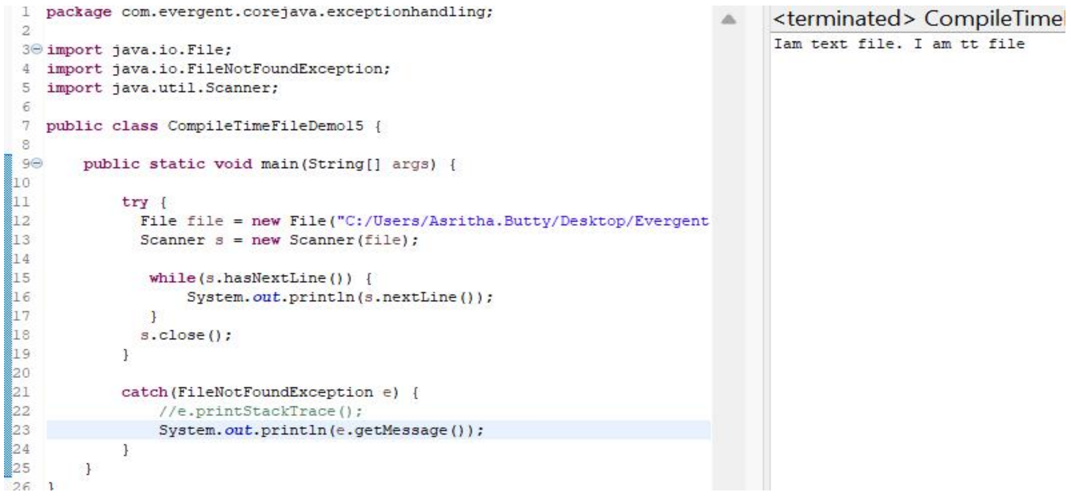
13. Points Covered :- ArrayIndexOutOfBoundsException



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

public class ArrayIndexOutOfBoundsException13 {

    public static void main(String[] args) {
        int numbers[] = {1,2,3,4,5};
        try {
            System.out.println("Accessing element at index 10 : "+numbers[10]);
        }

        catch(ArrayIndexOutOfBoundsException e) {
            System.out.println(e);
        }
        System.out.println("Program Continues");
    }
}
```

```
<terminated> ArrayIndexOutOfBoundsException13 [Java Application] (
java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 5
Program Continues
```

14. Points Covered :- FileNotFoundException

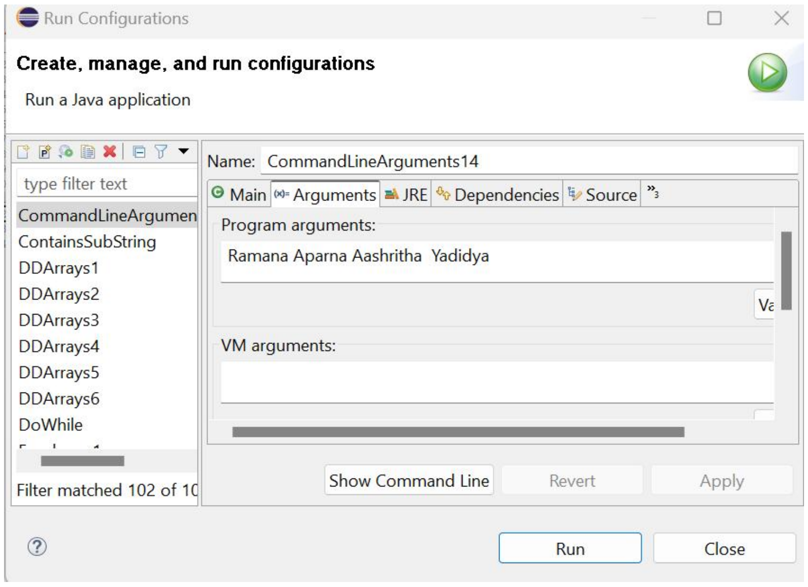


```
package com.evergent.corejava.exceptionhandling;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class CompileTimeFileDemo15 {

    public static void main(String[] args) {

        try {
          File file = new File("C:/Users/Asritha.Butty/Desktop/Evergent
          Scanner s = new Scanner(file);

           while(s.hasNextLine()) {
               System.out.println(s.nextLine());
           }
          s.close();
        }

        catch(FileNotFoundException e) {
            //e.printStackTrace();
            System.out.println(e.getMessage());
        }
    }
}
```

```
<terminated> CompileTime
Iam text file. I am tt file
```

Points Covered :- Command Line Arguments

Inserting Values using commandline arguments in string



Run Configurations

Create, manage, and run configurations

Run a Java application

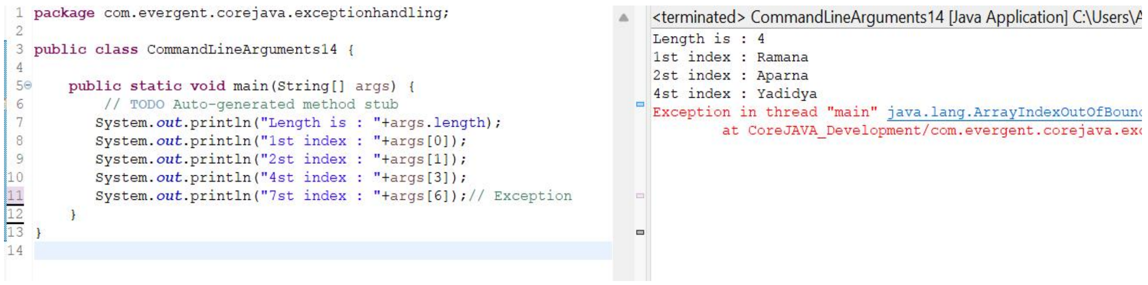
Name: CommandLineArguments14

Program arguments:

Ramana Aparna Aashritha Yadidya

VM arguments:

b) CommandLineArguments



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

public class CommandLineArguments14 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Length is : "+args.length);
        System.out.println("1st index : "+args[0]);
        System.out.println("2st index : "+args[1]);
        System.out.println("4st index : "+args[3]);
        System.out.println("7st index : "+args[6]);// Exception
    }
}
```

```
<terminated> CommandLineArguments14 [Java Application] C:\Users\A
Length is : 4
1st index : Ramana
2st index : Aparna
4st index : Yadidya
Exception in thread "main" java.lang.ArrayIndexOutOfBoun
        at CoreJAVA_Development/com.evergent.corejava.ex
```

Java Beans :-

Java bean is a mechanism

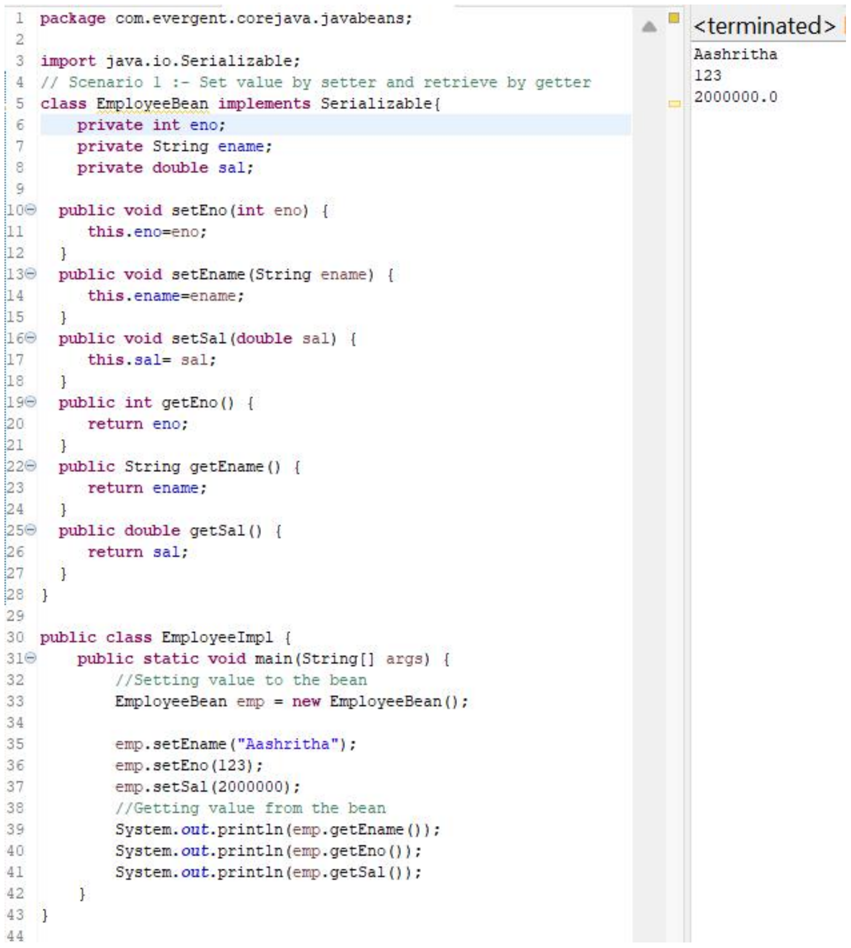
Java bean is light weight

All attributes are private and get/set methods are public.

Implements java.io.serializable Interface.

We can achieve tightly encapsulation through java beans.

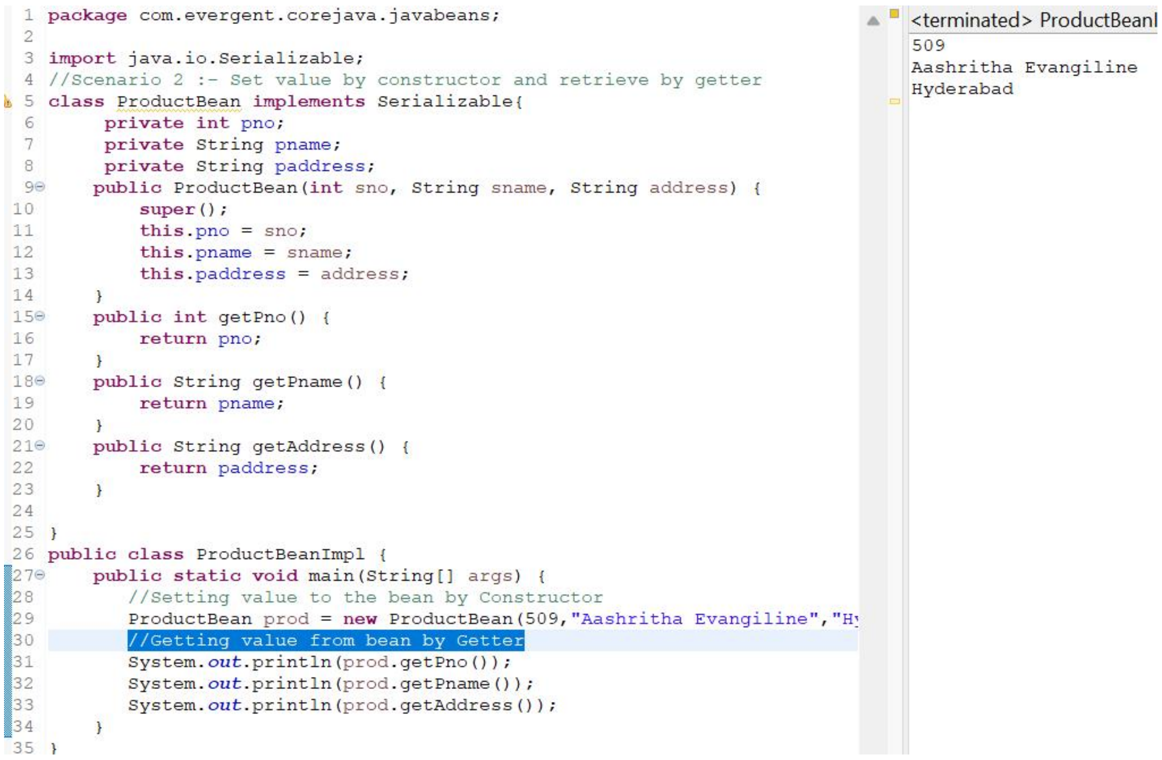
Points Covered :- Intiallizing by **setter** and getting by **getter**



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

import java.io.Serializable;
// Scenario 1 :- Set value by setter and retrieve by getter
class EmployeeBean implements Serializable{
    private int eno;
    private String ename;
    private double sal;

  public void setEno(int eno) {
     this.eno=eno;
  }
  public void setEname(String ename) {
     this.ename=ename;
  }
  public void setSal(double sal) {
     this.sal= sal;
  }
  public int getEno() {
     return eno;
  }
  public String getEname() {
     return ename;
  }
  public double getSal() {
     return sal;
  }
}

public class EmployeeImpl {
    public static void main(String[] args) {
        //Setting value to the bean
        EmployeeBean emp = new EmployeeBean();

        emp.setEname("Aashritha");
        emp.setEno(123);
        emp.setSal(2000000);
        //Getting value from the bean
        System.out.println(emp.getEname());
        System.out.println(emp.getEno());
        System.out.println(emp.getSal());
    }
}
```

```
<terminated>
Aashritha
123
2000000.0
```

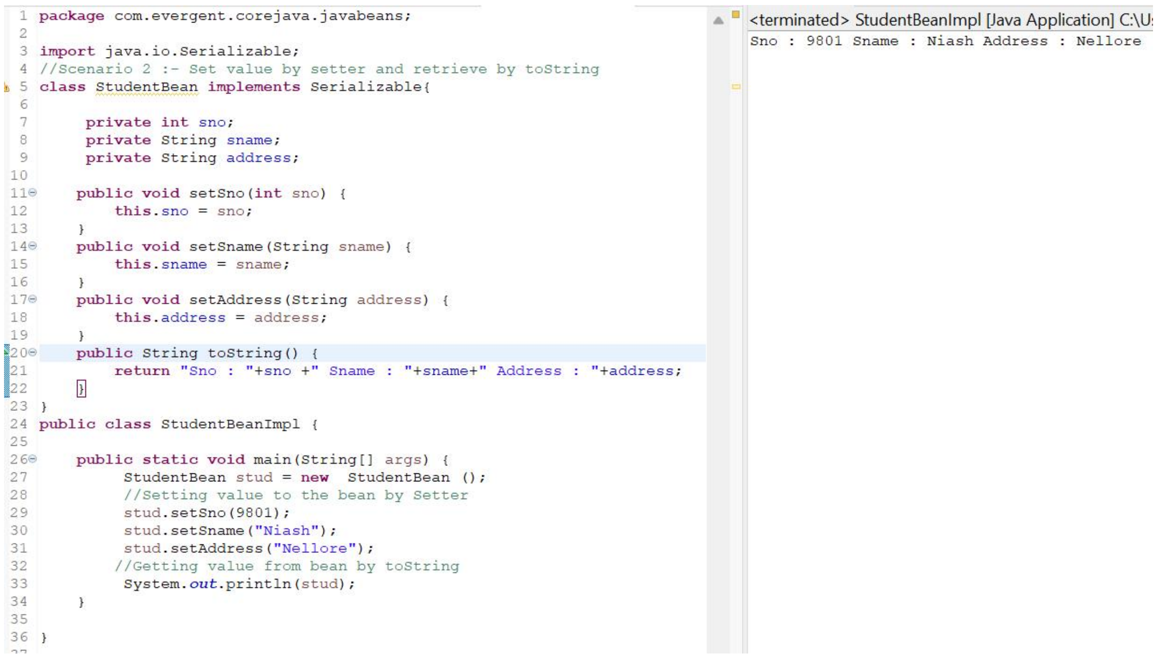
2. **Points Covered :-** Intiallizing by **Constructor** and getting by **getter**



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

import java.io.Serializable;
//Scenario 2 :- Set value by constructor and retrieve by getter
class ProductBean implements Serializable{
     private int pno;
     private String pname;
     private String paddress;
    public ProductBean(int sno, String sname, String address) {
        super();
        this.pno = sno;
        this.pname = sname;
        this.paddress = address;
    }
    public int getPno() {
        return pno;
    }
    public String getPname() {
        return pname;
    }
    public String getAddress() {
        return paddress;
    }

}
public class ProductBeanImpl {
    public static void main(String[] args) {
       //Setting value to the bean by Constructor
       ProductBean prod = new ProductBean(509,"Aashritha Evangiline","Hy
       //Getting value from bean by Getter
       System.out.println(prod.getPno());
       System.out.println(prod.getPname());
       System.out.println(prod.getAddress());
    }
}
```

```
<terminated> ProductBeanI
509
Aashritha Evangiline
Hyderabad
```

3. **Points Covered :-** Intiallizing by **Setter** and getting by **toString**

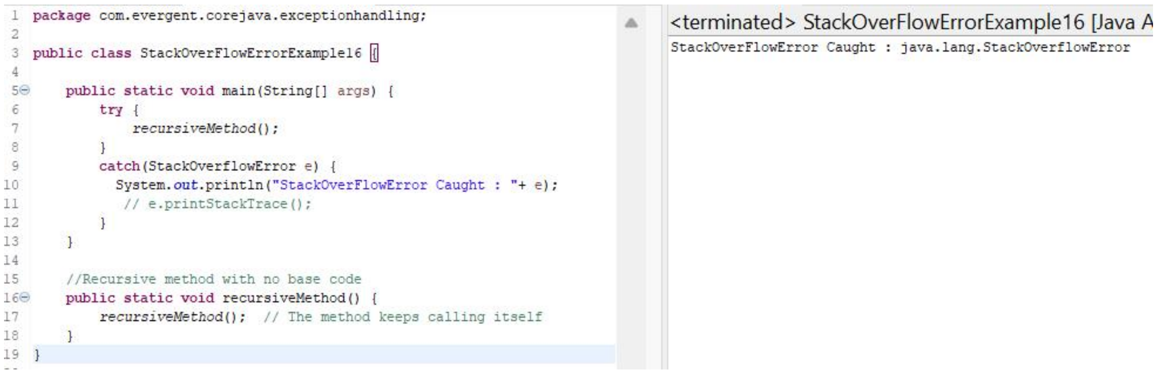


```java
package com.evergent.corejava.javabeans;

import java.io.Serializable;
//Scenario 2 :- Set value by setter and retrieve by toString
class StudentBean implements Serializable{

     private int sno;
     private String sname;
     private String address;

    public void setSno(int sno) {
        this.sno = sno;
    }
    public void setSname(String sname) {
        this.sname = sname;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String toString() {
        return "Sno : "+sno +" Sname : "+sname+" Address : "+address;
    }
}
public class StudentBeanImpl {

    public static void main(String[] args) {
         StudentBean stud = new  StudentBean ();
         //Setting value to the bean by Setter
         stud.setSno(9801);
         stud.setSname("Niash");
         stud.setAddress("Nellore");
        //Getting value from bean by toString
         System.out.println(stud);
    }

}
```

```
<terminated> StudentBeanImpl [Java Application] C:\U
Sno : 9801 Sname : Niash Address : Nellore
```

DAY-13 (22/08/2024)

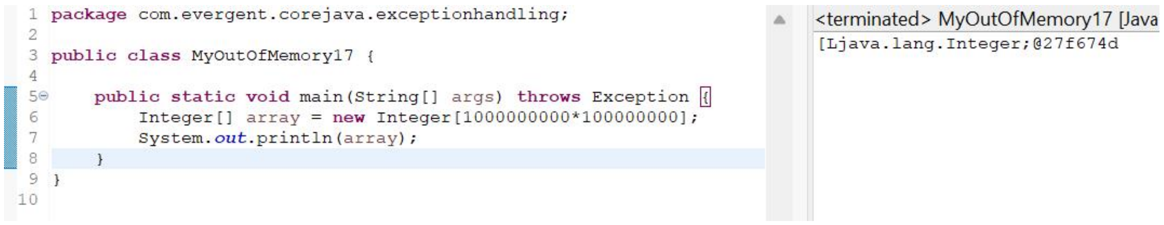
16. **Points Covered :-StackOverflowError**



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

public class StackOverFlowErrorExample16 {

    public static void main(String[] args) {
        try {
            recursiveMethod();
        }
        catch(StackOverflowError e) {
          System.out.println("StackOverFlowError Caught : "+ e);
           // e.printStackTrace();
        }
    }

    //Recursive method with no base code
    public static void recursiveMethod() {
        recursiveMethod();  // The method keeps calling itself
    }
}
```

```
<terminated> StackOverFlowErrorExample16 [Java A
StackOverFlowError Caught : java.lang.StackOverflowError
```

17. **Points Covered :- MyOutOfMemoryError**

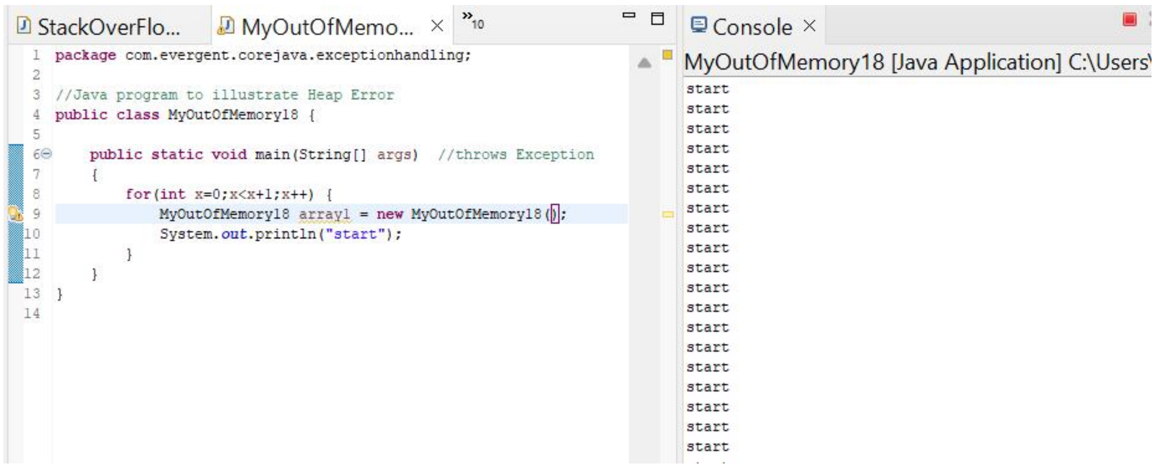


```java
package com.evergent.corejava.exceptionhandling;

public class MyOutOfMemory17 {

    public static void main(String[] args) throws Exception {
        Integer[] array = new Integer[1000000000*100000000];
        System.out.println(array);
    }
}
```

```
<terminated> MyOutOfMemory17 [Java
[Ljava.lang.Integer;@27f674d
```

17.2. Points Covered :- MyOutOfMemoryError // No modifications were accepted.

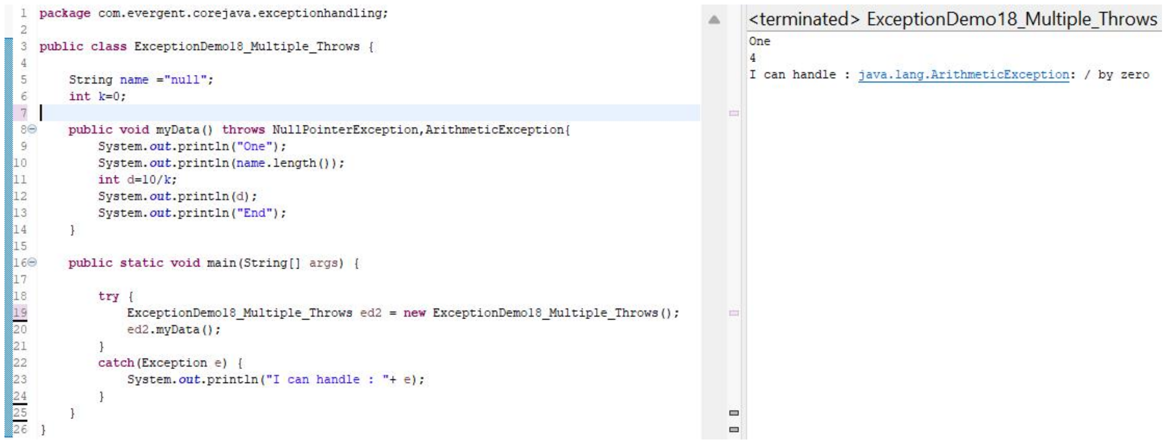
I have deleted the code



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

//Java program to illustrate Heap Error
public class MyOutOfMemory18 {

    public static void main(String[] args)  //throws Exception
    {
        for(int x=0;x<x+1;x++) {
            MyOutOfMemory18 array1 = new MyOutOfMemory18();
            System.out.println("start");
        }
    }
}
```

```
MyOutOfMemory18 [Java Application] C:\Users\
start
start
start
start
start
start
start
start
start
start
start
start
start
start
start
start
start
start
start
```

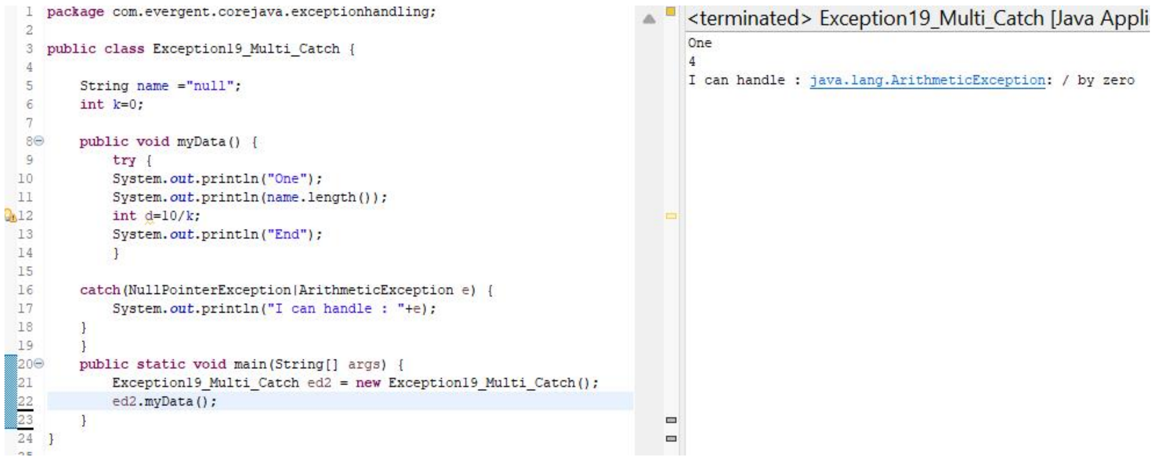
18. **Points Covered :- 23, multiple throws propagation**



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

public class ExceptionDemo18_Multiple_Throws {

    String name ="null";
    int k=0;

    public void myData() throws NullPointerException,ArithmeticException{
        System.out.println("One");
        System.out.println(name.length());
        int d=10/k;
        System.out.println(d);
        System.out.println("End");
    }

    public static void main(String[] args) {

        try {
            ExceptionDemo18_Multiple_Throws ed2 = new ExceptionDemo18_Multiple_Throws();
            ed2.myData();
        }
        catch(Exception e) {
            System.out.println("I can handle : "+ e);
        }
    }
}
```

```
<terminated> ExceptionDemo18_Multiple_Throws
One
4
I can handle : java.lang.ArithmeticException: / by zero
```

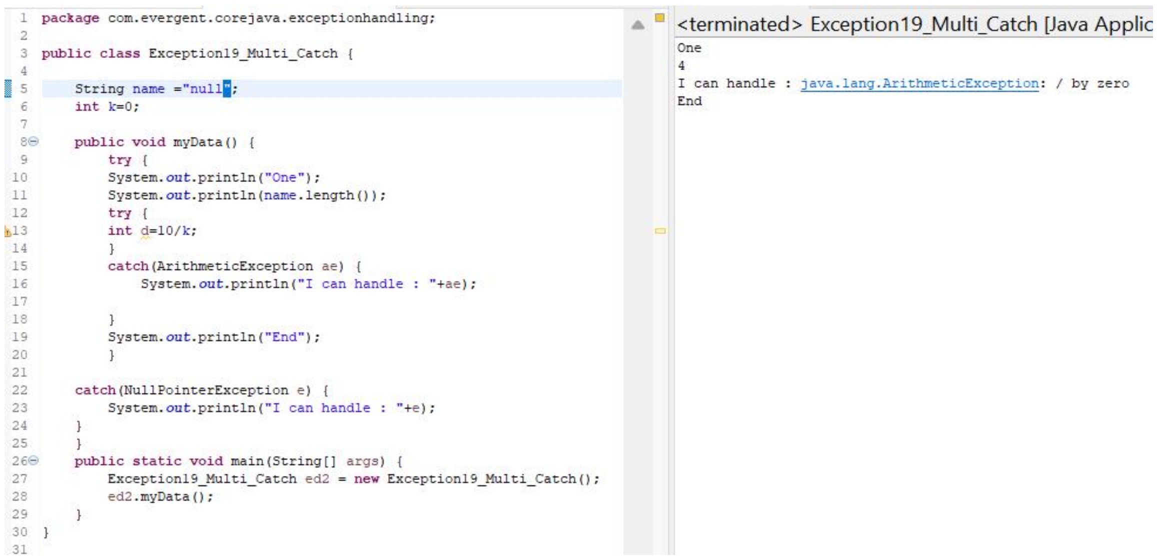
19. **Points Covered :- 24, multiple exceptions in one catch block**



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

public class Exception19_Multi_Catch {

    String name ="null";
    int k=0;

    public void myData() {
        try {
        System.out.println("One");
        System.out.println(name.length());
        int d=10/k;
        System.out.println("End");
        }

    catch(NullPointerException|ArithmeticException e) {
        System.out.println("I can handle : "+e);
    }
    }
    public static void main(String[] args) {
        Exception19_Multi_Catch ed2 = new Exception19_Multi_Catch();
        ed2.myData();
    }
}
```

```
<terminated> Exception19_Multi_Catch [Java Appli
One
4
I can handle : java.lang.ArithmeticException: / by zero
```

20. **Points Covered :- 25, Nested try-catch block**



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

public class Exception19_Multi_Catch {

    String name ="null";
    int k=0;

    public void myData() {
        try {
        System.out.println("One");
        System.out.println(name.length());
        try {
        int d=10/k;
        }
        catch(ArithmeticException ae) {
            System.out.println("I can handle : "+ae);

        }
        System.out.println("End");
        }

    catch(NullPointerException e) {
        System.out.println("I can handle : "+e);
    }
    }
    public static void main(String[] args) {
        Exception19_Multi_Catch ed2 = new Exception19_Multi_Catch();
        ed2.myData();
    }
}
```

<terminated> Exception19_Multi_Catch [Java Applic

```
One
4
I can handle : java.lang.ArithmeticException: / by zero
End
```