Core JAVA training - index

Date:05/08/2024

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

Date:06/08/2024
Morning(11:00 am)

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

After Noon(3:30 pm)

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

Date:07/08/2024
Mrng(11:00 am)

1.oops

- Encapsulation
  Programs
  Calculation
  Person
  Method Flow

After Noon(3:30 pm)

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

Date:08/08/24

Constructor

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

Program1:

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

public class Employee1 {
    Employee1(){
        System.out.println("default constructor");
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         new Employee1();

    }

}
```

```
<terminated> Employee1 [Java Application] C:\Users\rakshitha.banda\
default constructor
```

Program2:

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

public class Employee2 {
    int eno;
    String ename;
    double sal;
    public  Employee2()
    {
        System.out.println("default constructor...");
    }
    public Employee2(int eno1,String ename1,double sal1)
    {
        eno=eno1;
        ename=ename1;
        sal=sal1;
    }
    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) {
        Employee2 emp1=new Employee2();
        Employee2 emp2=new Employee2(123,"Shiva",60000);
        emp1.display();
        emp2.display();


    }

}
```

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

Program3:

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

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


    }

}
```

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

Program4:

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

public class Employee4 {

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

```
<terminated> Employee4 [Java Applica
Default constructor..
```

Program5:

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

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


    }
}
```

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

Program 6:

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

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

Program 7:

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

public class Car7 {
    String color;
    int maxSpeed;
    Car7()
    {
        color="White";
        maxSpeed=120;
    }
    Car7(String color,int maxSpeed)
    {
        this.color=color;
        this.maxSpeed=maxSpeed;
    }
    void display()
    {
        System.out.println("color:"+color);
        System.out.println("maxSpeed:"+maxSpeed);
    }
    public static void main(String[] args) {
        Car7 c1=new Car7();
        Car7 c2=new Car7("red",150);
        c1.display();
        c2.display();
    }

}
```

```
<terminated> Car7 [Java Applicat
color:White
maxSpeed:120
color:red
maxSpeed:150
```

Program8:

```java
class Animal
{
    private String name;
    private int age;
    public Animal(String name,int age)
    {
        this.name=name;
        this.age=age;
    }
    public void displayInfo() {
        System.out.println("Name:"+name);
        System.out.println("Age:"+age);
    }
}
class Dog extends Animal{
    private String breed;
    public Dog(String name,int age,String breed)
    {
        super(name,age);
        this.breed=breed;
    }
    public void displayInfo()
    {
        super.displayInfo();
        System.out.println("Breed:"+breed);
    }
}
public class Inheritance_Overriding8 {
    public static void main(String[] args) {
        Dog dog=new Dog("Buddy",5,"Golden Retriver");
        dog.displayInfo();
    }

}
```

```
<terminated> Inheritance_Overriding8 [J
Name:Buddy
Age:5
Breed:Golden Retriver
```

Program9:

```
package com.evelgent.corejava.constructor;

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

<terminated> Student9 [Java

```
Name:shiva
Age:20
Name:shiva
Age:20
```

Date:09/08/2024 - Day5

1. Static

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

Program1:

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

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

Program2:

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

public class StaticDemo2 {
    static String cname="India";
    static public void myData()
    {
        System.out.println("static method:myData");
    }
    public static void main(String[] args) {
        System.out.println(StaticDemo2.cname);
        StaticDemo2.myData();
    }
}
```

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

Program3:

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

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

}
```

```
<terminated> StaticDemo3 [J
my data..
```

Program4:

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

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

}
```

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

Program5:

```java
package com.evergent.corejava.static1;
//static block
public class staticDemo5 {
    static {
        System.out.println("static block");
    }
    static String name="India";
    static public void myData() {
        System.out.println("mydata");
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println(name);

    }

}
```

```
<terminated> staticDemo5 [Java Application] C:\Users\rakshitha.banda\D
static block
India
```

Program6:

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

    }

}
```

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

2. Final

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

Program1:

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

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

}
```

```
<terminated> FinalDem
India
```

Program2:

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

class MyClass{
    final public void myProducts()
    {
        System.out.println("AI products..");
    }
}
public class FinalDemo2 extends MyClass {

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

}
```

```
<terminated> Fi
India
```

Program3:

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

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

    final String cname="India";
    /*Cannot override the final method from MyCl
     public void myProducts()
    {
        System.out.println("AI products..");
    }*/
    public void myData()
    {
        System.out.println(cname);
    }
    public static void main(String[] args) {
        FinalDemo3 fd1=new FinalDemo3();
        fd1.myData();
        MyClass1 m1=new MyClass1();
        m1.myProducts();


    }

}
```

```
<terminated> FinalDemo3
India
AI products..
```

12/08/24 :

Strings:

-Why string is immutable?

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

StringBuffer:

a. String buffer is a final class

b. String buffer is mutable
c. String buffer having methods
d. All methods are synchronized

String Builder:

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

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

public class StringDemo1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str1=new String("Java");
        String str2=new String("Java");
        if(str1==str2)
            System.out.println("True");
        else
            System.out.println("False");


    }

}
```

```
<terminated> StringDemo1 [Java Application] C:\Users\rakshitha.banda\
False
```

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

public class StringDemo2 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        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> StringDemo2 [Java Application] C:\Users\rakshitha.banda\
True
True
```

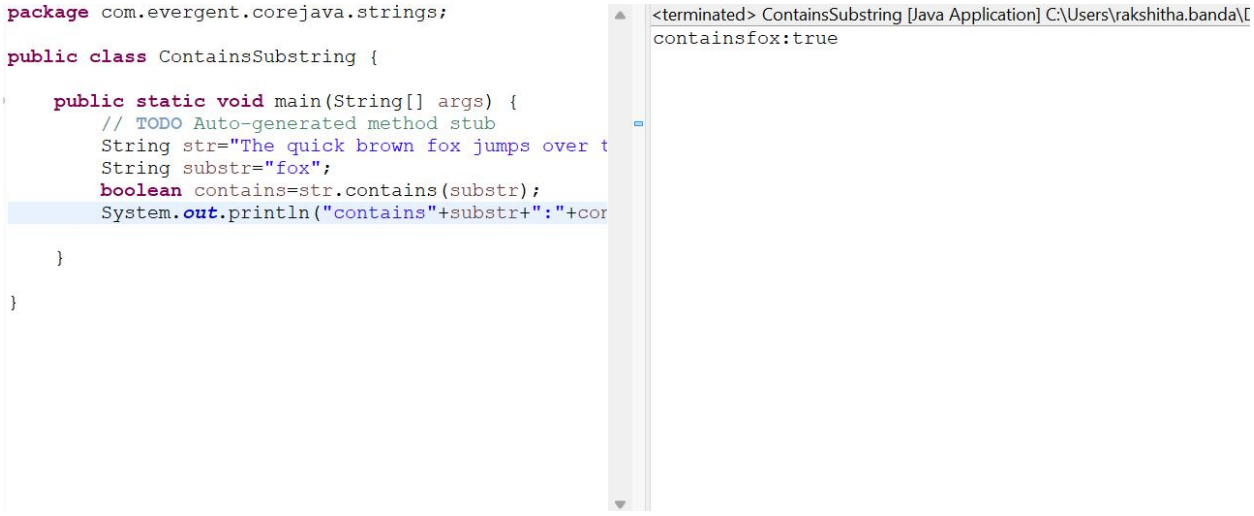
eclaration ×
evergent.corejava.static1.staticDemo5.java

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

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

public class StringDemo3_methods {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String name=new String("   JAVAWorld");
        System.out.println(name.length());
        System.out.println(name.toLowerCase());
        System.out.println(name.toUpperCase());
        System.out.println(name.trim()); // It will


    }

}
```

```
<terminated> StringDemo3_methods [Java Application] C:\Users\rakshitha.banda\
12
   javaworld
   JAVAWORLD
JAVAWorld
```

1.create a java program that creates a string and checks if it contains specific subString and then prints out the result



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

public class ContainsSubstring {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str="The quick brown fox jumps over t
        String substr="fox";
        boolean contains=str.contains(substr);
        System.out.println("contains"+substr+":"+cor

    }

}
```

```
<terminated> ContainsSubstring [Java Application] C:\Users\rakshitha.banda\[
containsfox:true
```

2. Write a java program to create a String ,remove all spaces from the string and then print out the result

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

public class RemoveSpaces {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str="Hello world, this is a test";
        String nospaces=str.replace(" ", "");
        System.out.println(nospaces);

    }

}
```

```
<terminated> RemoveSpaces [Java Application] C:\Users\rakshitha.banda\De
Helloworld,thisisatest
```

String concatination:

Strings can be concatenated using + operator (or) concat.

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

public class String_Concat {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str="Hello";
        str=str.concat(" World");
        System.out.println(str);

    }

}
```

```
<terminated> String_Concat [Java Application] C:\Users\rakshitha.banda\D
Hello World
```

Reverse of a String:

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

public class ReverseString {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str="Hello World!";
        StringBuilder reversed=new StringBuilder(str
        System.out.println(reversed);


    }

}
```

```
<terminated> ReverseString [Java Application] C:\Users\rakshitha.banda\
!dlroW olleH
```

Splitting of any sentence (or) text on spaces

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

public class SplitDemo1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str="Java is a powerful programming l
        //split the string by space
        String[] words=str.split(" ");
        for(int i=0;i<words.length;i++) {
            System.out.println(words[i]);
        }

    }

}
```

```
<terminated> SplitDemo1 [Java Application] C:\Users\rakshitha.banda\De
Java
is
a
powerful
programming
language
```

eclaration ×

Using for each loop:

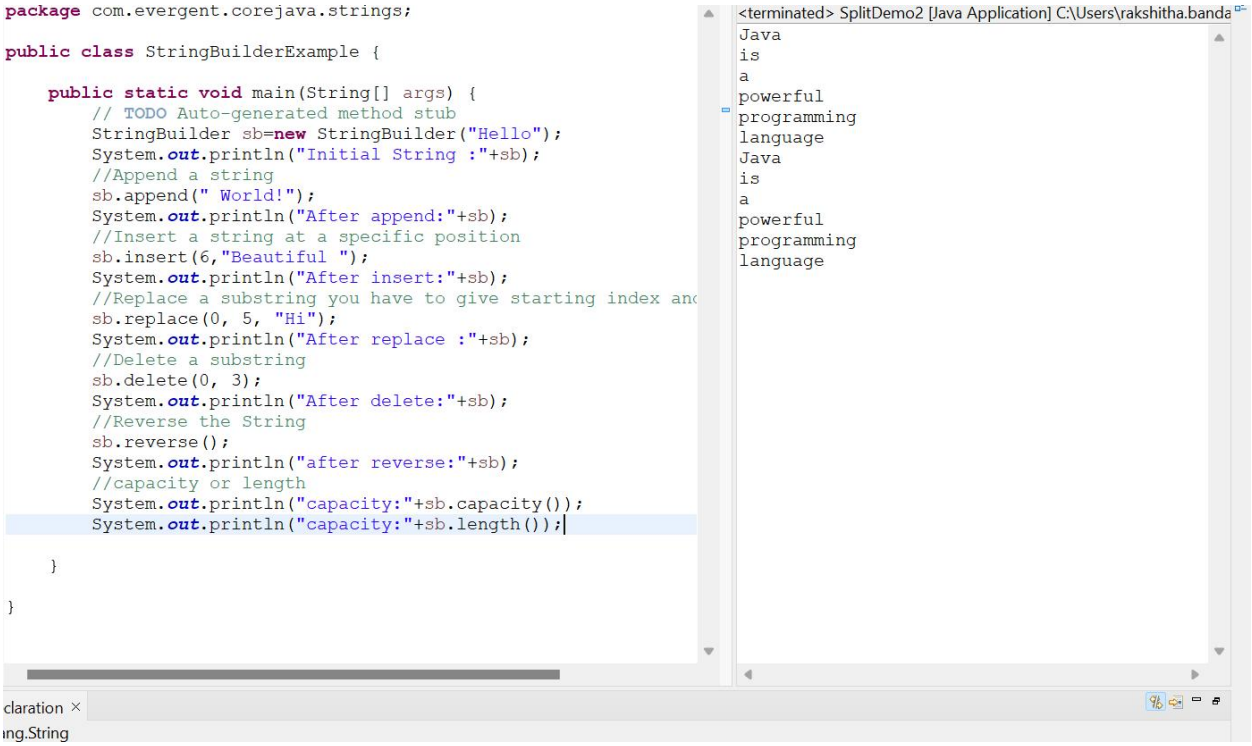
```java
package com.evergent.corejava.strings;

public class SplitDemo2 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str="Java is a powerful programming l
        //split the string by space
        String[] words=str.split(" ");
        for(int i=0;i<words.length;i++) {
            System.out.println(words[i]);
        }
        for(String w:words) {
            System.out.println(w);
        }
    }

}
```

```
<terminated> SplitDemo2 [Java Application] C:\Users\rakshitha.banda\Deskto
Java
is
a
powerful
programming
language
Java
is
a
powerful
programming
language
```

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

public class StringBufferExample {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        StringBuffer sb=new StringBuffer("Hello");
        System.out.println("Initial String :"+sb);
        //Append a string
        sb.append(" World!");
        System.out.println("After append:"+sb);
        //Insert a string at a specific position
        sb.insert(6,"Beautiful ");
        System.out.println("After insert:"+sb);
        //Replace a substring you have to give starting index and
        sb.replace(0, 5, "Hi");
        System.out.println("After replace :"+sb);
        //Delete a substring
        sb.delete(0, 3);
        System.out.println("After delete:"+sb);
        //Reverse the String
        sb.reverse();
        System.out.println("after reverse:"+sb);
        //capacity or length
        System.out.println("capacity:"+sb.capacity());
        System.out.println("capacity:"+sb.length());



    }

}
```

```
<terminated> SplitDemo2 [Java Application] C:\Users\rakshitha.banda
Java
is
a
powerful
programming
language
Java
is
a
powerful
programming
language
```

eclaration ×



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

public class StringBuilderExample {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        StringBuilder sb=new StringBuilder("Hello");
        System.out.println("Initial String :"+sb);
        //Append a string
        sb.append(" World!");
        System.out.println("After append:"+sb);
        //Insert a string at a specific position
        sb.insert(6,"Beautiful ");
        System.out.println("After insert:"+sb);
        //Replace a substring you have to give starting index and
        sb.replace(0, 5, "Hi");
        System.out.println("After replace :"+sb);
        //Delete a substring
        sb.delete(0, 3);
        System.out.println("After delete:"+sb);
        //Reverse the String
        sb.reverse();
        System.out.println("after reverse:"+sb);
        //capacity or length
        System.out.println("capacity:"+sb.capacity());
        System.out.println("capacity:"+sb.length());

    }

}
```

```
<terminated> SplitDemo2 [Java Application] C:\Users\rakshitha.banda
Java
is
a
powerful
programming
language
Java
is
a
powerful
programming
language
```


String class important points:

1. In java a string is a sequence of characters ,often used to represent text.
2. Strings are objects in java and are instances of the string class,which is part of the java java.lang package
3. Key features of strings in java:
   A. Immutable:once a string object is created ,it cannot be changed .
4. Java optimizes memory usage by storing strings in special area of memory as string pool
5. If two strings have the same value and are created without using new keyword they will refer to same object in the stringpool.
6. We can create a string in java in multiple ways:
   a. Using string literals :str=”hello world”;
   b. Using the new keyword
      String str=new String(“hello, world”);

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

public class StringPerformance1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String a;
        String b;
        System.out.println('a'+'b');
        System.out.println('a'+3);

    }

}
```

```
<terminated> StringPerformance1 [Java Application] C:\Users\raksh
195
100
```

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

public class StringPerformance2 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String a;
        String b;
        System.out.println((char) ('a'+3));
        System.out.println("a"+"b");

    }

}
```

```
<terminated> StringPerformance2 [Java Application] C:\Users\rakshith
d
ab
```

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

public class StringPerformance3 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String series="";
        for(int i=0;i<26;i++) {
            char ch=(char)('a'+i);
            series+=ch;
        }
        System.out.println(series);

    }

}
```

```
<terminated> StringPerformance3 [Java Application] C:\Users\rakshith
abcdefghijklmnopqrstuvwxyz
```

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

public class StringPerformance4 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        StringBuilder builder=new StringBuilder();
        for(int i=0;i<26;i++) {
            char ch=(char)('a'+i);
            builder.append(ch);
        }
        System.out.println(builder);

    }

}
```

```
<terminated> StringPerformance4 [Java Application] C:\Users\rakshith
abcdefghijklmnopqrstuvwxyz
```

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

import java.util.Arrays;

public class StringPerformance5 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
    String name="JavaTechnologies";
    String str=Arrays.toString(name.toCharArray());
    System.out.println(str);
    System.out.println(name.indexOf('T'));
    System.out.println("    Java       ".strip());
    System.out.println("    Java       ".strip().length());


    }

}
```

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

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

public class PersonImmutable {
        // TODO Auto-generated method stub
        private final String name;
        private final int age;

        //constructor to initialize the final fields
        public PersonImmutable(String name,int age) {
            this.name=name;
            this.age=age;

        }
        public String myName() {
            //name="chinni";
            return name;

        }
        public int myAge() {
            return age;
        }

    }
```

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

public class MainPerson {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        PersonImmutable person=new PersonImmutable("rakshitha",20);
        System.out.println("Name:"+person.myName());
        System.out.println("Age:"+person.myAge());


    }

}
```

```
<terminated> MainPerson [Java Application] C:\Users\rakshitha.ba
Name:rakshitha
Age:20
```

Object Class Methods:

I) toString()

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

 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 class MyData {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ImmutableString str=new ImmutableString("Hello World !");
        System.out.println(str.myValue());
        System.out.println(str.toString());
        System.out.println(str);

    }

}
```

```
<terminated> MyData [Java Application] C:\Users\rakshitha.banda\
Hello World !
Hello World !
Hello World !
```

ii)hashCode():

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

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


public class MyPerson {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Person p1=new Person("rakshitha",22);
        System.out.println(p1);
        System.out.println(p1.hashCode());

    }

}
```

```
<terminated> MyPerson [Java Application] C:\Users\rakshitha.banda
Name:rakshithaAge:22
212628335
```

eclaration ×

Interfaces

1. Interface is a keyword
2. We can declare method signature only but not implementations
3. By default all interface methods are abstract
4. If any class implementations interface that class should be override all interface methods otherwise that class will be showing compile time error
5. We can not create object to interface but we can create reference to interface
6. We can declare variables inside interface by default public static final
7. Java will support multiple inheritance through interfaces
8. One class can implements more than one interfaces
9. One interface extend other interfaces
10. Interface with zero methods (without method) is called marker interface
11. Marker interface are java.io.serializable.clonable

Book Interface Program:

```
package com.evergent.corejava.interface1;

public interface Book {
    String cName="India";
    public void bookTitle();
    public void bookAuthor();
    public void bookPrice();

}
```

Program1:

```java
package com.evergent.corejava.interface1;

public class BookImpl implements Book {
    public void bookTitle() {
        System.out.println("Core java");
    }
    public void bookAuthor() {
        System.out.println("oracle crop:"+cName);
    }
    public void bookPrice() {
        System.out.println("Rs.550");
    }
    public void show() {
        System.out.println("local methods of BookImpl");
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        BookImpl book1=new BookImpl();
        book1.bookAuthor();
        book1.bookTitle();
        book1.bookPrice();
        book1.show();

    }

}
```

```
<terminated> BookImpl [Java Application] C:\Users\rakshitha.banda
oracle crop:India
Core java
Rs.550
local methods of BookImpl
```

Program2:

```java
package com.evergent.corejava.interface1;

public class BookImpl implements Book {
    public void bookTitle() {
        System.out.println("Core java");
    }
    public void bookAuthor() {
        System.out.println("oracle crop:"+cName);
    }
    public void bookPrice() {
        System.out.println("Rs.550");
    }
    public void show() {
        System.out.println("local methods of BookImpl");
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        BookImpl book1=new BookImpl();
        book1.bookAuthor();
        book1.bookTitle();
        book1.bookPrice();
        book1.show();

    }

}
```

```
<terminated> BookImpl [Java Application] C:\Users\rakshitha.banda
oracle crop:India
Core java
Rs.550
local methods of BookImpl
```

MyNewData Interface:

```
package com.evergent.corejava.interface1;

public interface MyNewData {
    public void dataInfo();
}
```

NewBook Interface:

```
package com.evergent.corejava.interface1;

public interface NewBook extends MyNewData{
    public void mynewBook();
    public void bookPrice();

}
```

Program3:

```java
package com.evergent.corejava.interface1;

public class BookImpl3 implements Book,NewBook{

    public void bookTitle() {
        System.out.println("Core java");
    }
    public void bookAuthor() {
        System.out.println("oracle crop:"+cName);
    }
    public void bookPrice() {
        System.out.println("Rs.550");
    }
    public void mynewBook() {
        System.out.println("my new book");
    }
    public void dataInfo() {
        System.out.println("my data info");
    }
    public void show() {
        System.out.println("local methods of BookImpl");
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        BookImpl3 book1=new BookImpl3();
        book1.bookAuthor();
        book1.bookTitle();
        book1.bookPrice();
        book1.show();
        book1.mynewBook();
        book1.dataInfo();

    }

}
```

```
<terminated> BookImpl3 [Java Application] C:\Users\rakshitha.ban
oracle crop:India
Core java
Rs.550
local methods of BookImpl
my new book
my data info
```

Abstract Class:

1. Abstract is a Keyword
2. Abstract class having abstract methods 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 not create object to abstract class but we can create object reference to abstract class
6. We can declare abstract class if it has zero abstract methods for security purpose because we can not create object to it
7. We can create construtcor to abstarct class
8. We can access abstract class constructor through sub class object creation

Product Abstract Class:

```
package com.evergent.corejava.abstract1;

abstract public class Product {
         String cName="rakshitha";
        abstract public void newProduct();
        public void allProducts() {
            System.out.println("All products");
        }

    }
```

Program1:

```java
package com.evergent.corejava.abstract1;
//if you will not override the abstract method you have to make the
//because it internally consist of abstract method
//abstract public class ProductImpl extends Product{
public class ProductImpl extends Product{
    public void newProduct() {
        System.out.println("My new Product");
    }
    public void show() {
        System.out.println("Local methods of product Impl class");
    }
    public static void main(String[] args) {
        ProductImpl p1=new ProductImpl();
        p1.show();
        p1.newProduct();
        p1.allProducts();

    }

}
```

```
<terminated> ProductImpl [Java Application] C:\Users\rakshitha.ba
Local methods of product Impl class
My new Product
All products
```

Program2:

```java
package com.evergent.corejava.abstract1;

public class ProductImpl2 extends Product {

    public void newProduct() {
        cName="welcome";
        //by default variables in abstract classes are not final
        System.out.println("My new Product:"+cName);
    }
    public void show() {
        System.out.println("Local methods of product Impl class");
    }
    public static void main(String[] args) {
        //can not instantiate the type product
        //Product p1=new Product();
        Product p2=new ProductImpl2();
        p2.allProducts();
        p2.newProduct();
        //p2.show();

    }

}
```

```
<terminated> ProductImpl2 [Java Application] C:\Users\rakshitha.ba
All products
My new Product:welcome
```

Program3:

Product3 abstract class:

```
package com.evergent.corejava.abstract1;
 //we can create constructor for abstract classes
abstract public class Product3 {
    public Product3() {
        System.out.println("Product3 abstract class constructor");
    }
    String cName="India";
    abstract public void newProduct();
    public void allProducts() {
        System.out.println("ALl products");
    }

}
```

```java
package com.evergent.corejava.abstract1;

public class ProductImpl3 extends Product3{

	public ProductImpl3() {
		System.out.println("ProductImpl3 sub class Constructor"
	}
	public void newProduct() {
		cName="welcome";
		System.out.println("My new Product:"+cName);
	}
	public void show() {
		System.out.println("Local methods of product Impl class
	}
	public static void main(String[] args) {
		//can not instantiate the type product
		//Product p1=new Product();
		ProductImpl3 p2=new ProductImpl3();
		p2.allProducts();
		p2.newProduct();
		p2.show();

    }

}
```

```
<terminated> ProductImpl3 [Java Application] C:\Users\rakshitha.ba
Product3 abstract class constructor
ProductImpl3 sub class Constructor
ALl products
My new Product:welcome
Local methods of product Impl class
```

Exception Handling:

1. Exception handling is a mechanism
2. Exceptions are in-built mechanism of java
3. All exceptions are executed while abnormal conditions only
4. Normal flow it won't execute exceptions
5. Once any exceptions are occuring in java then remaining lines of code is unreachable
6. Java.lang.throwable is super class for Exception and Errors
7. There are two types of Exceptions in java
    A.checked exception
    B.Unchecked exception

8.All checked exceptions are compile-time exceptions
9.All unchecked exceptions are runtime exceptions
10.There are 5 keywords in exception handling
    A.try
    B.catch()
    C.finally()
    D.throws
    E.throw

11.try is for business logic
12.catch is for handling exceptions
13.finally is block,if exception is occur or not finally block is executed
14.throws an exception will be executed method by method
15.throw is for runtime exception and will call predefined or user-defined exception
16.try followed by either catch block or finally block
17.we should follow exceptions hierarchy
18.we can create our own(user defined) exception
19.Our own exceptions extends exception or runtime exception
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 can not handle errors

Program1:

```java
package com.evergent.corejava.exceptionhandling;
 //Exception handling is a mechanism
//Exceptions are in built mechanism of java
public class ExceptionDemo1 {

    public static void main(String[] args) {
        //String name="rakshitha";
        String name=null;
        System.out.println(name.length());

    }

}
```

```
<terminated> ExceptionDemo1 [Java Application] C:\Users\rakshitha
Exception in thread "main" java.lang.NullPo
        at com.evergent.corejava.exceptionh
```

Program2:

```java
package com.evergent.corejava.exceptionhandling;
//All Exceptions are excecuted while abnormal condition only
//Normal flow it won't execute any exceptions
public class ExceptionDemo2 {
    String name=null;
    public void myData() {
        try {
            System.out.println("one");
            System.out.println(name.length());
            System.out.println("end");
        }
    catch(NullPointerException e) {
        System.out.println("I acn handle :"+e);
    }

    }

    public static void main(String[] args) {
        ExceptionDemo2 ed=new ExceptionDemo2();
        ed.myData();

    }

}
```

```
<terminated> ExceptionDemo2 [Java Application] C:\Users\rakshitha.
one
I acn handle :java.lang.NullPointerExceptio
```

Program3:

```
package com.evergent.corejava.exceptionhandling;
 //there is two exceptions in class,
//developers should handle one after one
public class ExceptionDemo3 {

        String name="null";
        int k=0;
        public void myData() {
            try {
                System.out.println("one");
                System.out.println(name.length());
                int t=10/k;
                System.out.println(t);
                System.out.println("end");
            }
        catch(NullPointerException e) {
            System.out.println("I acn handle :"+e);
            }
            catch(ArithmeticException e) {
                System.out.println("I can handle :"+e);
            }

        }
    public static void main(String[] args) {
        ExceptionDemo3 ed=new ExceptionDemo3();
        ed.myData();

    }

}
```

```
<terminated> ExceptionDemo3 [Java Application] C:\Users\rakshitha
one
4
I can handle :java.lang.ArithmeticException
```

Program4:

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

//exception should follow exception heirarchial
public class ExceptionDemo4 {

    String name=null;
    int k=0;
    public void myData() {
        try {
            System.out.println("one");
            System.out.println(name.length());
            int t=10/k;
            System.out.println(t);
            System.out.println("end");
        }
    catch(NullPointerException e) {
        System.out.println("I acn handle :"+e);
        }
        catch(ArithmeticException e) {
            System.out.println("I can handle :"+e);
        }
        catch(Exception e) {
            System.out.println("I can handle :"+e);
        }

    }
public static void main(String[] args) {
    ExceptionDemo3 ed=new ExceptionDemo3();
    ed.myData();



    }
}
```

```
<terminated> ExceptionDemo4 [Java Application] C:\Users\rakshitha.
one
4
I can handle :java.lang.ArithmeticException
```

Program5:

```
package com.evergent.corejava.exceptionhandling;
//finally is block,if exceptions is occur not finally block is excec
public class ExceptionDemo5final1 {
        String name=null;
        int k=2;
        public void myData() {
            try {
                System.out.println("one");
                System.out.println(name.length());
                int t=10/k;
                System.out.println(t);
                System.out.println("end");
            }
        catch(NullPointerException e) {
            System.out.println("I acn handle :"+e);
            }
            catch(ArithmeticException e) {
                System.out.println("I can handle :"+e);
            }
            catch(Exception e) {
                System.out.println("I can handle :"+e);
            }
            finally {
                System.out.println("finally block close connection")
            }

        }
    public static void main(String[] args) {
        ExceptionDemo5final1 ed=new ExceptionDemo5final1();
        ed.myData();




    }

}
```

```
<terminated> ExceptionDemo5final1 [Java Application] C:\Users\raks
one
I acn handle :java.lang.NullPointerExceptio
finally block close connection
```

Program6:

```
package com.evergent.corejava.exceptionhandling;
//try followed by either catch block or finally block
public class ExceptionDemo6final2 {

    String name="null";
    public void myData() {
        try {
            System.out.println("one");
            System.out.println(name.length());
            System.out.println("end");
        }
    finally{
        System.out.println("finally block is close");
    }

    }

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

    }

}
```

```
<terminated> ExceptionDemo6final2 [Java Application] C:\Users\raks
one
4
end
finally block is close
```

Program7:

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

//throws an exception will be executed method by method
public class ExceptionDemo7throws1 {
    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 {
            ExceptionDemo7throws1 ex1=new ExceptionDemo7throws1()
            ex1.myData();
        }
        catch(Exception e) {
            System.out.println("i can handle :"+e);
        }

    }

}
```

```
<terminated> ExceptionDemo7throws1 [Java Application] C:\Users\rakshi
one
i can handle :java.lang.NullPointerException:
```

Program8:

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

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("mychange method");
        }

        public static void main(String[] args) {
            try {
                ExceptionDemo8throws2 ex1=new ExceptionDemo8throws2()
                ex1.myData();
            }
            catch(Exception e) {
                System.out.println("i can handle :"+e);
            }

    }

}
```

```
<terminated> ExceptionDemo8throws2 [Java Application] C:\Users\
one
i can handle :java.lang.NullPointerExcepti
```

Program9:

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

class ProductNotFoundException extends Exception{
    public ProductNotFoundException(String mes) {
        System.out.println("hello "+mes);
    }
}
public class ProductImpl {
    int pno=105;
    public void myData() throws ProductNotFoundException{
        if(pno>100) {
            throw new ProductNotFoundException("there is no product")
        }
        else {
            System.out.println("product are there");
        }
    }

    public static void main(String[] args) {
        try {
            ProductImpl pro=new ProductImpl();
            pro.myData();
        }
        catch(ProductNotFoundException e) {
            System.out.println(e);
            System.out.println(e.getMessage());
        e.printStackTrace();
        }

    }

}
```

```
<terminated> ProductImpl (1) [Java Application] C:\Users\rakshitha
hello there is no product
com.evergent.corejava.exceptionhandling.Pr
null
com.evergent.corejava.exceptionhandling.Pr
        at com.evergent.corejava.exception
        at com.evergent.corejava.exception
```

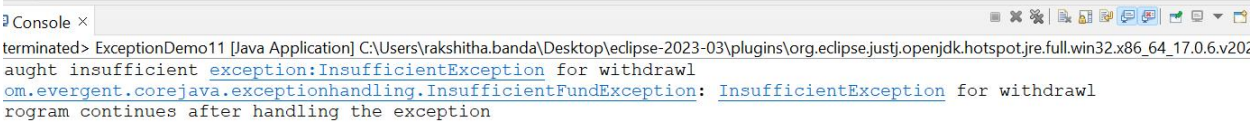
Program10:

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

class AgeNotSupportException extends Exception{
    public AgeNotSupportException(String msg) {
        System.out.println(msg);
    }
}
public class AgeNotSupport {
    int age=10;
    public void myAge() throws AgeNotSupportException{
        if(age<18) {
            throw new AgeNotSupportException("age is below 18");
        }
        else {
            System.out.println("age is supported for voter card");
        }
    }

    public static void main(String[] args) {
        try {
        AgeNotSupport a=new AgeNotSupport();
        a.myAge();}
        catch(AgeNotSupportException e) {
            System.out.println(e);
        }

    }

}
```

```
<terminated> AgeNotSupport [Java Application] C:\Users\rakshitha
age is below 18
com.evergent.corejava.exceptionhandling.Ag
```

Program11:

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


class InsufficientFundException extends Exception{
    public InsufficientFundException(String message) {
        //System.out.println(message);
        super(message);
    }
}
public class ExceptionDemo11 {
    //method that throws a custom checked exception
    public static void withdraw(double amount)throws InsufficientFundException{
        double balance=500.00;
        if(amount>balance) {
            throw new InsufficientFundException("InsufficientException for withdrawl");
        }
        else {
            System.out.println("withdrawl succesful");
        }}
    public static void main(String[] args) {
        try {
            withdraw(600.00);//this will trigger insufficient exception

        }
        catch(InsufficientFundException e) {
            System.out.println("caught insufficient exception:"+e.getMessage());
            System.out.println(e);
            System.out.println("program continues after handling the exception");
        }
```



Program12:

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

class InvalidScoreException extends Exception{
    public InvalidScoreException(String message) {
        //System.out.println(message);
        super(message);
    }
}
public class ExceptionDemo12 {
    //method that throws a custom checked exception
    public static void validateScore(int score)throws InvalidScoreException{

        if(score<0 ||score>100) {
            throw new InvalidScoreException("score must be between 0 and 100 ");
        }
        else {
            System.out.println("withdrawl succesful");
        }}
    public static void main(String[] args) {
        try {
            validateScore(600);//this will trigger insufficient exception

        }
        catch(InvalidScoreException e) {
            System.out.println("caught insufficient exception:"+e.getMessage());
            System.out.println(e);
            System.out.println("program continues after handling the exception");
        }




    }
```

Console ×

<terminated> ExceptionDemo12 [Java Application] C:\Users\rakshitha.banda\Desktop\eclipse-2023-03\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32

```
caught insufficient exception:score must be between 0 and 100
com.evergent.corejava.exceptionhandling.InvalidScoreException: score must be between 0 and 100
program continues after handling the exception
```

Program13:

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

public class ExceptionDemo13 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
 int[] arr= {1,2,3,4};
    try {
     System.out.println(arr[10]);
    }
    catch(ArrayIndexOutOfBoundsException e) {
     System.out.println(e.getMessage());
     System.out.println(e);
    }
    }

}
```

<terminated> ExceptionDemo13 [Java Application] C:\Users\rakshitha.banda\Desktop\e

```
Index 10 out of bounds for length 4
java.lang.ArrayIndexOutOfBoundsException: Index 10 out of
```

Program14:

```java
package com.evergent.corejava.exceptionhandling;
//commandline exception
public class ExceptionDemo14 {

    public static void main(String[] args) {
        System.out.println(args[0]);
        System.out.println(args[1]);
        System.out.println(args[2]);


    }

}
```

```
<terminated> ExceptionDemo14 [Java Application] C:\Users\rakshitha.banda\Desktop\ec
1
2
Exception in thread "main" java.lang.ArrayIndexOutOfBounds
        at com.evergent.corejava.exceptionhandling.Exceptio
```

Program15:

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

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
public class ExceptionDemo15 {

    public static void main(String[] args) {
        try {
            //attempt to open a file does not exist
            File file=new File("c:/myData/myinfo.txt")
            Scanner sc=new Scanner(file);
            while(sc.hasNextLine()) {
                System.out.println(sc.nextLine());
            }
            sc.close();
        }
        catch(FileNotFoundException e) {
            //handle the filenotfoundexception
            //System.out.println("file not found excep
            e.printStackTrace();
        }

    }

}
```

```
<terminated> ExceptionDemo15 [Java Application] C:\Users\rakshitha.banc
my name is rakshitha
```

Stackoverflow Error
Program16:

```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:"
        }

    }
    public static void recursiveMethod() {
        recursiveMethod();
    }

}
```

```
<terminated> StackOverFlowErrorExample16 [Java Application] C:\Users\rakshitha.ba
stackoverflowerror caught:null
```

OutofMemory Error
Program17:

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

public class MyOutofMemory {

    public static void main(String[] args) {
        Integer[] array=new Integer[100000000*100000000];
        try {
        System.out.println(array);}
        catch(OutOfMemoryError e) {
            System.out.println("hi"+e);
        }
    }

}
```

```
<terminated> MyOutofMemory [Java Application] C:\Users\rakshitha.banda\Deskto
Exception in thread "main" java.lang.OutOfMemoryError
        at com.evergent.corejava.exceptionhandling.My(
```

JavaBeans:

1. JavaBean is a mechanism
2. JavaBean is a lightweight
3. All the attributes are private and getters and setters are public implements java.io.serializable interface
4. we can achieve tightly encapsulation through javabeans

Program1:

```
package com.evergent.corejava.javabeans;
import java.io.Serializable;
public class Employee implements Serializable{
    private int eno;
    private String ename;
    private double sal;

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


}
```

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

public class EmployeeImpl {

    public static void main(String[] args) {
        Employee emp=new Employee();
        //initializing values through setter methods
        emp.setEno(100);
        emp.setEname("rakshitha");
        emp.setSal(10000);
        //getting values from getter methods

    System.out.println("Employee no:"+emp.getEno());
    System.out.println("Employee name:"+emp.getEname());
    System.out.println("Employee salary:"+emp.getSal());



    }

}
```

```
<terminated> EmployeeImpl (1) [Java Application] C:\Users\rakshitha.banda
Employee no:100
Employee name:rakshitha
Employee salary:10000.0
```

Program2:

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

import java.io.Serializable;
public 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;
    }



}
```

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

public class ProductImpl {

    public static void main(String[] args) {
        //initializing values through constructor
        Product product=new Product(10,"laptop",85000);
        //getting values through getter methods

System.out.println("product n0 :"+product.getPno());
System.out.println("Product name:"+product.getPname());
System.out.println("Product price:"+product.getPrice());



    }

}
```

```
<terminated> ProductImpl (2) [Java Application] C:\Users\rakshitha.ban
product n0 :10
Product name:laptop
Product price:85000.0
```

Program3:

```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+"\n student address:"+address;
    }



}
```

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

public class StudentImpl {

    public static void main(String[] args) {
        //initializing values through setter methods
        StudentBean stu=new StudentBean();
        stu.setSno(10);
        stu.setSname("rakshitha");
        stu.setAddress("nizamabad");

        //getting values through toString method

        System.out.println(stu);
    }

}
```

```
<terminated> StudentImpl [Java Application] C:\Users\rakshitha.banda\De
student no:10
 student name:rakshitha
 student address:nizamabad
```