CORE JAVA PROGRAMMING LANGUAGE

OOPS (OBJECT ORIENTED PROGRAMING LANGUAGES)

- PYTHONE
- C++
- JAVA

FEATURES:

- **Encapsulation**
- **Inheritance**
- **Abstraction**
- **Polymorphism**

What does object oriented programming means?

If any programming language follow me these four features in that language is called object oriented programming languages. And those features are Encapsulation ,Inheritance ,Abstraction ,Polymorphism.

Encapsulation

- ✓ Definition
- ✓ Binding of data into a single entity

In java why it is required?

By default it is always there.

```
package com.mycompany.priyanshurajaccedmy;

public class car {

   private int speed;

   public void addspeed (int sp){

      if (sp>500){

      speed=sp;

      }
```

```
}
public int givespeed(){
return speed;
}
}
```

```
package com.mycompany.priyanshurajaccedmy;
public class carTest {
   public static void main(String[] args) {
       car c=new car();
      //c.speed=-900;
      c.addspeed (-45);
      System.out.println(c.givespeed());

   }

}
```

Why do make global variable private?

Because it not private then anyone can add any vaiue. 0 into it to restrict we made it private and access through methords those methords may be getter and setters.

Inheritance

- ✓ Definition
- ✓ If a properties from parent class coming to child class then it is called inheritance.

Why ?

Reusability

We use extends keywords in this.

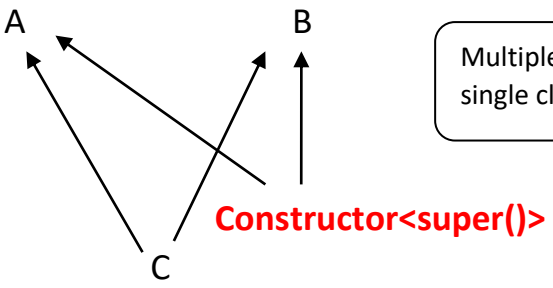
Type of inheritance

1. Single
2. Multilevel
3. Multiple
4. Hierarchal
5. Hybrid

- **Single Inheritance**
  One class is inherit another class.
- **Multilevel inheritance**
  One class is inherit another class and another class is inherit another class.
- **Multiple inheritance**
  There are multiple class is super class but one class as a sub class and it is not allowed java.
- **Hierarchal inheritance**
  One super class and many sub class and again many sub class.

Why multiple class not allowed in java ?

- Super()
- If same method which will inherited



Multiple inheritance means multiple super classes and single class it is not supported in java.

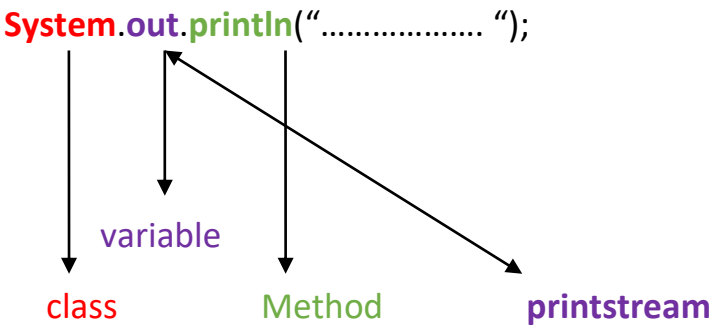
Acess specifiers /Acess modifiers

Acess specifiers

- Private ⟶ In java we can have variable and methord as a private which cannot be access outside.
- Protected ⟶ Within in same package and from a subclass of different package.
- Default ⟶ Default members are accessible within same package.
- Public ⟶ Access everywhere.

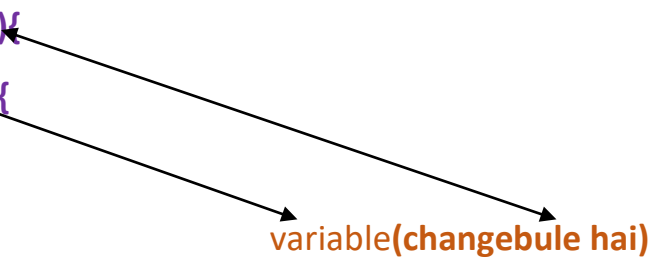
Static

- Single copy storage.
- It is used for memory management. It applied to variable /methords.
- Some examples
  >>>>math.abs( )-inbuilt
  >>>>>intger.min-value



static public void main (string[]args){

public static void main (string[]run){



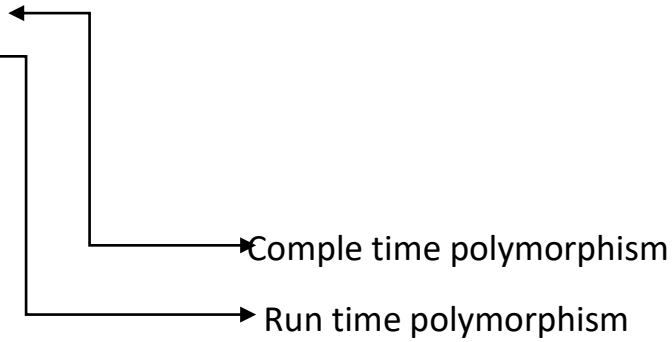
In dono condition mai ye run karga …….

static public main (string[]args){ ⟷ void hatne kay karan run nahi karegar

public static void main (string run){ ⟷ [ ] hatne kay karan run nahi karega

Polymorphism

- Different behavior at different time by same entites.
- Itcan achived 2 ways
  1. Methord overloading
  2. Methord overriding



Method overloading:

1. Method name should be same.
2. Parameters should be different.
3. Acess specifies can be different.
4. Return type can be anything.

Method overriding:

1. Method name must be same.
2. Parameters must be same.
3. Acess specifies must be same or higher than super class method.
4. Return type must be same.
5. Super class sub class should be there.

Abstraction

- Exposing only required things is called abstraction.
- it can be achieved by using **interface** and **abstract class.**

Interface

- it is same as that of class except
  method- public abstract
  variable –public static final
- it cannot have constructor
- we cannot create object of interface

  - collection
  - serialization

Abstract Class

- To use this class we need to use inheritance
- It can have abstract and non abstract method
- If we want to use abstract method of this class then we must extends this class and override abstract method of super class/ implemented the method

  - Inputstream
  - Outputstream

Final

- Means constant cannot changed
- We can apply final to variable/method and class
- If any variable is final then cannot be changed
- If any method is final then cannot override
- If class is final then we cannot extend

Exception Handling

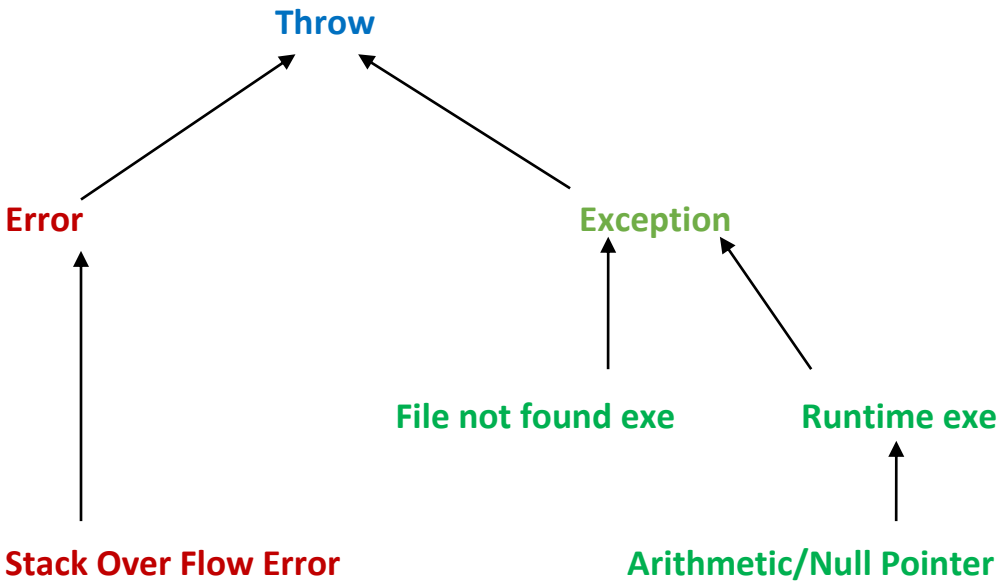
- To avoid termination of a program
- To achive that we have try,catch,finally,throw,throws

- Try > we need to write a code which may give exception
- Catch > to display error message
- Finally > this code will execute always even if exception occurs r not

Null pointer Arithmetic Exception……(Run time)
Class not found File not found……..(compile time)

Error/Exception

- **Error –** which cannot be handled
- **Exception-** which can be handled



Throw/Throws:- both are use to delegate and exception to caller of a methords

Final/Finally/Finalize

finalize():this is related to garbage collection concept , it is getting before garbage collection run.

Garbage collection: it is used for memory management we can suggest garbage collection .but it is no guaranteed .

System gc ( );- only suggestion can be given to JVM

Array

Int a [ ] = new [4];

```
package com.mycompany.priyanshurajaccedmy;
public class x {
   public static void main(String[] args) {
      int a[]=new int[4];

      for(int i=0;i<a.length;i++){
       System.out.println(i);

      }
   }

}
```

```
For (   ;  ;  )   {
……..
        }
```

Compulsory only 2 semicolon
Not error
Print infinite (unlimted times)

- **It is a collection of element with same data type is know as array**.

Array – it has utility method to help us to do operation with array element.

```
package com.mycompany.priyanshurajaccedmy;

public class z {

   public static void main(String[] args) {

      int a[]=new int[3];

      a[0]=22;

      a[1]=45;

      a[2]=90;

      System.out.println("one dimenssional array element are");

      System.out.println(a[0]);

      System.out.println(a[1]);

      System.out.println(a[2]);

   }

   }
```

Two dimensional array:

```
class TwodimensionalStandard
{
        public static void main(String args[])
        {
        int[][] a={{10,20},{30,40}};//declaration and initialization
        System.out.println("Two dimensional array elements are");
        System.out.println(a[0][0]);
        System.out.println(a[0][1]);
        System.out.println(a[1][0]);
        System.out.println(a[1][1]);
        }
}
```

Collection

Same as array

- It is same as that of array
- This class is in utill package
- It is used storing elements
- We can mention generic<> to have type safety
- To remove elements we can use removed method
- Allow duplicate
- Order maintained

```
package com.mycompany.priyanshurajaccedmy;

import java.util.array list;

public class R {

   public static void main(String[] args) {

      array list <integer>ai=new array list<>();

      al.add(45);

      al.add(45);

      al.add(45);

      al.add(45);

      al.add(45);


      for (int x:al){

         System.out.println(x);

      }

     }

}
```

Hash Map

```
package com.mycompany.priyanshurajaccedmy;

import java.util.array list;

public class R {

   public static void main(String[] args) {

    hashmap <integer>hm=new hashmap<integer,string>();

     hm.put(23,"india");

     hm.put(11,"india1");

     hm.put(34,"india2");

     hm.put(22,"india3");

     hm.put(11,"india4");

     hm.put(34,"india4");

         System.out.println(hm);

         System.out.println(hm.size());

     }

         }
```

Key is unique.

- It is exist in utill package
- In hash map is a classic java
- In hash map we have we c.value
- Duplicate key not allows
- To use and check a method to use size method
- To add a element to use add method

List	Set	Map
Array list	Hash set	Hash Map

Vector	Tree set	Tree Map
Linked list	linked Hash Set	Linked Hash Map
		Hash Table

Hash Set

- Utill
- Collection framework
- Duplicate not allowed
- Insertion order is not maintained
- To add element we use add method

```
package com.mycompany.priyanshurajaccedmy;
import java.util.HashMap;
import java.util.Set;
public class mn {
   public static void main(String[] args) {
      HashMap<Integer,String> hm= new HashMap <Integer, String>();

      hm.put(23, "india");
      hm.put(11, "india1");
      hm.put(34, "india2");
      hm.put(22, "india3");
      hm.put(11, "india4");
      hm.put(34, "india4");

      Set <Integer>keys=hm.keySet();
      for(int key:keys){
         System.out.println(key);
         String val =hm.get(key);
         System.out.println(val);
      }
   }
   }
```

String

- It is in language package
- String can be declared as string x=”kiran”
- String xx = new string (“jdk”);
- To perform operation on string we have equals upper case/ concat etc.
- It uses string pool concept

String Buffer

- All method of string buffer are synchronized
- Its class language package
- We use append method to add string in string buffer

String Builder

- **JDK 1.5**
- Same as that of String Buffer except method are non synchronized