Assignment-04

CSA0992-

Programming In JAVA for Freshers

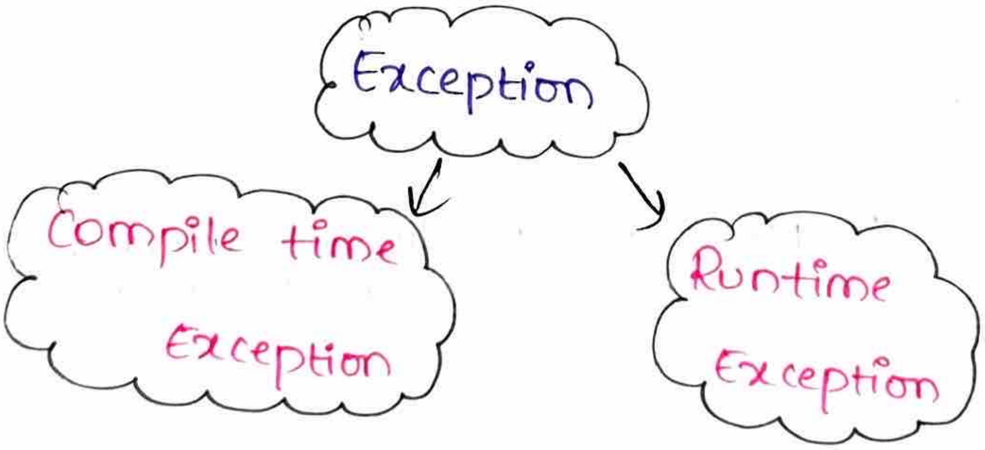
Name : K.V. Sai Sanjana
Reg.No : 192011124
Dept : CSE

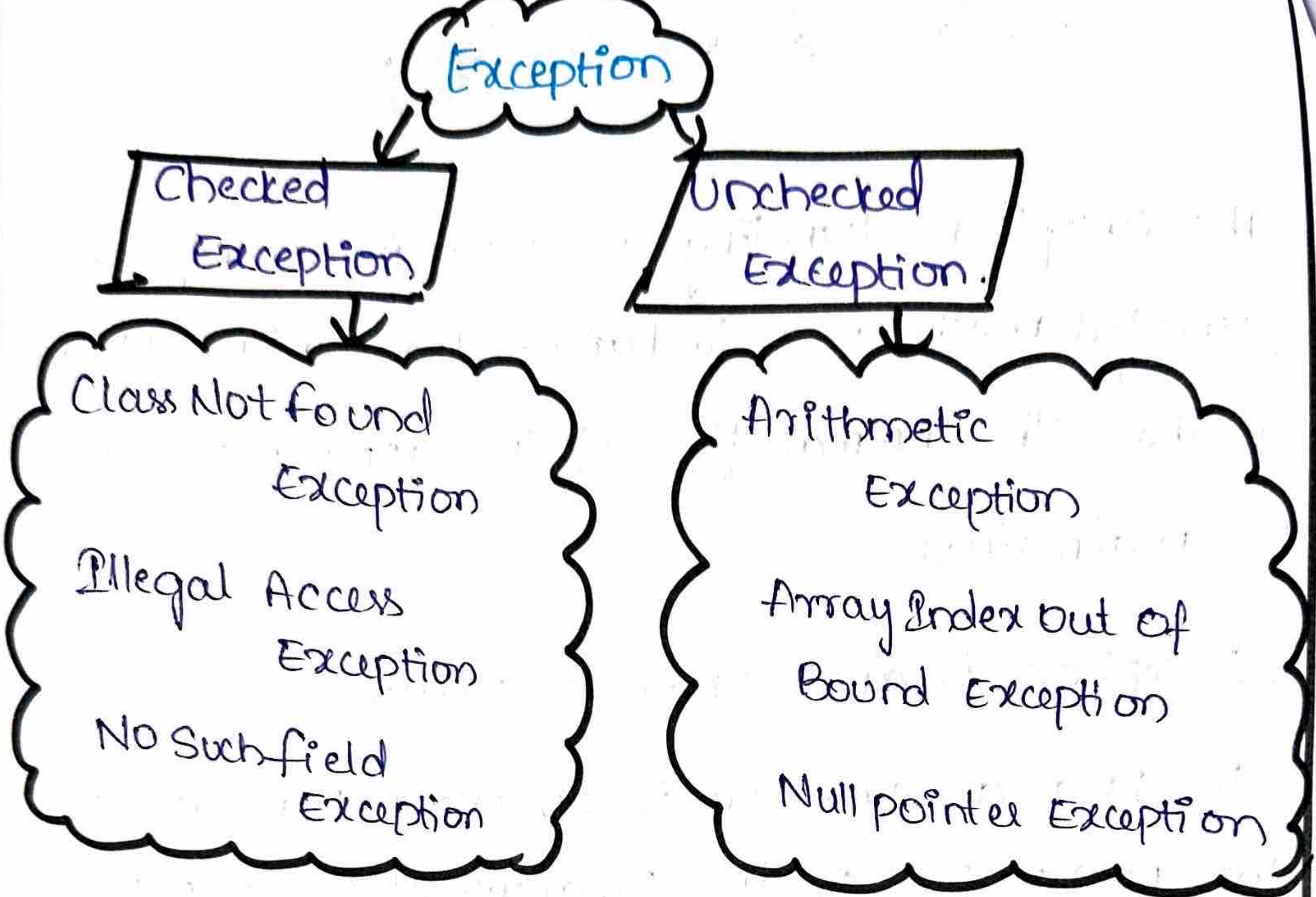
Exception Handling :-

* The exception Handling in Java is one of the powerful mechanism to handle the runtime errors so that the normal flow of the application can be maintained.

* Exception is an abnormal condition.
* In Java, an exception is an event that disrupts the normal flow of the program. It is an Object which is thrown at runtime.

* Exception Handling is a mechanism to handle runtime errors such as ClassNotFound Exception, IOException, SQLException, RemoteException, etc.





BUILT IN TYPE EXCEPTION:-

* Predefined Exception
* All types of checked and unchecked exception.

TYPES OF ERROR:-

Syntax Error:-

→ Due to poor understanding of language.

Logic Error:-

→ Poor understanding of problem.

JAVA EXCEPTION KEYWORDS:-

TRY:-

* The "Try" keyword is used to specify a block where we should place an exception code.
* It means we can't use try block alone.
* The try block must be followed by either catch or finally

Catch:-

* It is used to handle the exception.

finally:-

* The "finally" block is used to excecute the necessary code of the program.

Throw:-

* The "throw" keyword is used to throw an exception

Throws:-

* The "throws" keyword is used to declare exceptions.
* It specifies there are may occur an exception in the method.
* It doesn't throw an exception.
* It is always used with method signature.

General form of an exception-handling block

```
try {
// block of code to monitor for errors
}
catch (ExceptionType1 exOb) {
// exception handler for exceptionType1
}
catch (ExceptionType2 exOb) {
// exception handler for exception type 2
}
finally {
// block of code to be executed after try block ends
}
```

Example :-

```
Public class Java Exception {
  public static void main (String [] args) {
try {
  int data =100/0;
  }
catch (arithmetic Exception e)
{
System.out.println (e);
} System.out.Println("rest of the code...");
}
}
```

Example-2:-

```
import java.util.Random;
class HandleError {
   Public static void main (String args [])
   {
      int a=0 ,b=0 ,c=0;
      Random r= new Random();
      for (int i=0; i< 32000; i++)
      {
         try {
            b= r.nextInt ();
            c =r.nextInt();
            a = 12345/ (b/c);
         }
         catch (Arithmetic Exception e)
         {
          System.out.Println ("Division by zero");
            a =0;
         }
      System.out.println("a: " +a);
   }
}}
```

Multiple Catch Clauses Example

```
class Multiple catches {
  public static void main (String args[ ]) {
 try {
   int a = args. length;
   System. out · Println ("a ="+a );
   int b= 42 / a
   int c[ ]= {1};
    c[ 42] = 99;
  }
 catch (Arithmetic Exception e) {
    System. out. Println ("Divide by 0:" +e);
catch (Array Index out of BoundException e)
 {
   System. out. Println ("Array index oob:"+e);
 }
 System. Out . println ("After try/catch blocks.");
 }
}
```