EXCEPTION HANDLING

EXCEPTION - THE ERROR THAT OCCUR DURING THE RUNTIME OF THE PROGRAM AND IT WILL HANDLES BY THE PROGRAMMER USING EXCEPTION HANDLING.

TRY - THE TRY STATEMENT ALLOW US DEFINE A BLOCK OF CODE TO BE TESTED FOR ERRORS WHILE IT IS BEING EXECUTED.

CATCH - THE CATCH STATEMENT ALLOW TO DEFINE A BLOCK OF CODE TO BE EXECUTED,IF AN ERROR OCCUR IN THE TRY BLOCK.

EXAMPLE:

```
public class Main {
public static void main(String[ ] args) {
try {
int[] myNumbers = {1, 2, 3};
System.out.println(myNumbers[10]);
} catch (Exception e) {
System.out.println("Something went wrong.");
    }
  }
}
```

OUTPUT:

Something went wrong.

FINALLY - THE FINALLY STATEMENT LETS YOU EXECUTE CODE,AFTER TRY AND CATCH,REGARDLESS OF THE RESULT.

EXAMPLE:

```
public class Main {
public static void main(String[] args) {
try {
```

```
int[] myNumbers = {1, 2, 3};
System.out.println(myNumbers[10]);
} catch (Exception e) {
System.out.println("Something went wrong.");
} finally {
System.out.println("The 'try catch' is finished.");
    }
  }
}
```

OUTPUT:

Something went wrong.
The ‘try catch ‘ is finished.

THROW - THE THROW STATEMENT ALLOW YOU TO CREATE A CUSTOM ERROR AND IT IS USED TOGETTHER WITH AN EXCEPTION TYPE.

THERE ARE MANY EXCEPTION TYPES IN JAVA:

1) ArithematicException.
2) FileNotFoundException.
3) ArrayIndexOutOfBoundsException.
4) SecurityException.

EXAMPLE:

```
public class Main {
static void checkAge(int age) {
If (age < 18) {
throw new ArithmeticException("Access denied -you
      Must need 18 years to access);
 }
 else {
 System.out.println("Access granted - You are old
       enough!");
   }
 }
```

```
public static void main(String[] args) {
checkAge(13);
   }
  }
```

OUTPUT:

Acess Denied - you need atleast 18 years to access.

THREADS

THREAD - THREAD ALLOW A PROGRAM TO OPERATE MORE EFFICIENTLY BY DOING MULTIPLE THINGS AT THE SAME TIME AND IT IS USED TO PERFORM COMPLICATED TASKS IN THE BACKGROUND WITHOUT INTERRUPTING THE MAIN PROGRAM.

CREATING THREADS :

THERE ARE TWO WAYS TO CREATE A THREAD.

1) CREATING BY EXTENDING THE THREAD CLASS AND OVERRIDING ITS run() METHOD.

EXAMPLE:

```
public class Main extends Thread {
public void run() {
System.out.println("This code is running in a thread");
   }
 }
```

2) CREATING A THREAD BY IMPLEMENT THE Runnable INTERFACE.

EXAMPLE:

```
public class Main implements Runnable {
public void run() {
```

```
        System.out.println("This code is running in a thread");
          }
        }
```

IF THE CLASS IMPLEMENTS THE Runnable INTERFACE,THE THREAD CAN BE PASSING AN INSTANCE OF THE CLASS TO A Thread AND THEN CALLING THE THREAD start() METHOD.

EXAMPLE:

```
        public class Work {
        public static void main(String[] args) throws Exception {
        A a = new A();
         a.start();
        B b = new B();
        Thread th = new Thread(b);
        th.start();
         }
          }
        class A extends Thread {
         public void run() {
         for (int i = 0; i < 10; i++) {
          System.out.println("----a---" + i);
      try {
           Thread.sleep(1000);
      } catch (InterruptedException e) {

      }
    }
  }
}
class B implements Runnable{
  public void run() {
    for (int i = 0; i < 10; i++) {
      System.err.println("----b---" + i);
```

```
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {

            }
        }
    }
}
class C{
 }
```