**3rd OBJECTIVE:**

Extend stack with push and pop method with the help of stack class with an Exception handling capability, by raising an exception of type EmptyStackException whenever you try to print from an empty stack. Use the try-catch block when invoking the printStack method. When trying to print from an empty stack the message “Reading from an Empty Stack” should be displayed. You should Define the class EmptyStackException that derives from Exception.

**PROGRAM # 3:**

SOURCE CODE:

import java.util.EmptyStackException;

import java.util.Stack;

public class stack {

public static void main(String args[]){

Stack stck=new Stack();

try{

stck.push(10);

stck.push(20);

stck.push(45);

stck.pop();

stck.pop();

stck.pop();

stck.pop();

}

catch(EmptyStackException e){

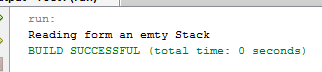
System.out.println("Reading form an emty Stack");

}

}

}

OUTPUT:



CONCLUSION:

In this program we are learning how to use stack methods.

**4th OBJECTIVE:**

**PROGRAM # 4:**

Create a class with a main( ) that throws an object of class Exception inside a try block. Give the constructor for Exception a String argument. Catch the exception inside a catch clause and print the String argument. Add a finally clause and print a message to prove you were there.

SOURCE CODE:

import java.io.FileNotFoundException;

public class myexception

{

myexception(String a)

{

a = "My exception";

System.out.println(a);

}

public static void main(String[] args) throws Exception

{

try

{

throw new FileNotFoundException();

}

catch (FileNotFoundException e)

{

throw new Exception("file not found!!!");

}

catch (Exception e)

{

System.out.println(e.getMessage());

}

finally

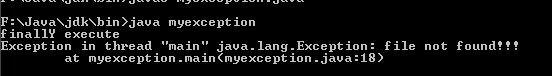
{

System.out.println("finallY execute");

}

}}

OUTPUT:



CONCLUSION:

In this program we are learning how to handle exceptions.

**5th OBJECTIVE:**

Create your own exception class using the extends keyword. Write a constructor for this class that takes a String argument and stores it inside the object with a String reference. Write a method that prints out the stored String. Create a try-catch clause to exercise your new exception.

**PROGRAM # 5:**

SOURCE CODE:

class myexception1 extends Exception

{

myexception1(String s)

{

super(s);

} }

class excep1

{ void disp(int a) throws myexception1

{ System.out.println("value ="+a);

throw new myexception1("Exception Occured!");

}

}

class excepmain

{

public static void main(String args[])

{

excep1 obj=new excep1();

try

{ obj.disp(10);

}

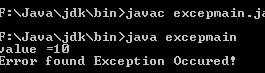
catch(myexception1 d)

{

System.out.println("Error found "+d.getMessage());

} }}

OUTPUT:



CONCLUSION:

In this program we are learning how to create own exception.