Java exception program :

package customexception;  
  
public class MailValidatorUtil {  
 public void validateMail(String mailId) throws InvalidMailException{  
 //is mail contain @  
 boolean isContainAt = mailId.contains("@");  
 boolean isContainDot = mailId.contains(".");  
  
 if(isContainAt && isContainDot){  
 System.*out*.println("Mail is valid");  
 }else {  
 System.*out*.println("Mail is invalid. Raising exception");  
 String message = "Mail " + mailId + " is not a valid mail";  
 InvalidMailException invalidMailException = new InvalidMailException(message);  
 //syntax of throw is "throw <exception object>"  
 throw invalidMailException;  
 }  
 }  
}

package customexception;  
  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.util.Scanner;  
  
public class MailValidatorDemo {  
 public static void main(String[] args){  
 MailValidatorUtil mailValidatorUtil = new MailValidatorUtil();  
  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter mail id");  
 String mail = sc.next();  
  
 try {  
 int result = 1/0;  
 mailValidatorUtil.validateMail(mail);  
 }catch (InvalidMailException t){  
 System.*out*.println("Exception : "+t.getMessage());  
 }catch (ArithmeticException t){  
 System.*out*.println("Arithematic Exception !!");  
 }catch (Exception e){  
 System.*out*.println("Generic Exception !!");  
 }  
 finally {  
 System.*out*.println("inside finally block");  
 }  
  
  
 String returnValue = *testExceptionReturnValue*();  
 System.*out*.println(returnValue);  
 }  
  
 private static String testExceptionReturnValue() throws RuntimeException{  
 try{  
 //throw new RuntimeException();  
 //return "returning from try";  
 System.*exit*(100);  
 }catch (Exception e){  
 return "returning from catch";  
 }finally {  
 return "returning from finally";  
 }  
 }  
}

package customexception;  
  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.util.Scanner;  
  
public class MailValidatorDemo {  
 public static void main(String[] args){  
 MailValidatorUtil mailValidatorUtil = new MailValidatorUtil();  
  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter mail id");  
 String mail = sc.next();  
  
 try {  
 int result = 1/0;  
 mailValidatorUtil.validateMail(mail);  
 }catch (InvalidMailException t){  
 System.*out*.println("Exception : "+t.getMessage());  
 }catch (ArithmeticException t){  
 System.*out*.println("Arithematic Exception !!");  
 }catch (Exception e){  
 System.*out*.println("Generic Exception !!");  
 }  
 finally {  
 System.*out*.println("inside finally block");  
 }  
  
  
 String returnValue = *testExceptionReturnValue*();  
 System.*out*.println(returnValue);  
 }  
  
 private static String testExceptionReturnValue() throws RuntimeException{  
 try{  
 //throw new RuntimeException();  
 //return "returning from try";  
 System.*exit*(100);  
 }catch (Exception e){  
 return "returning from catch";  
 }finally {  
 return "returning from finally";  
 }  
 }  
}