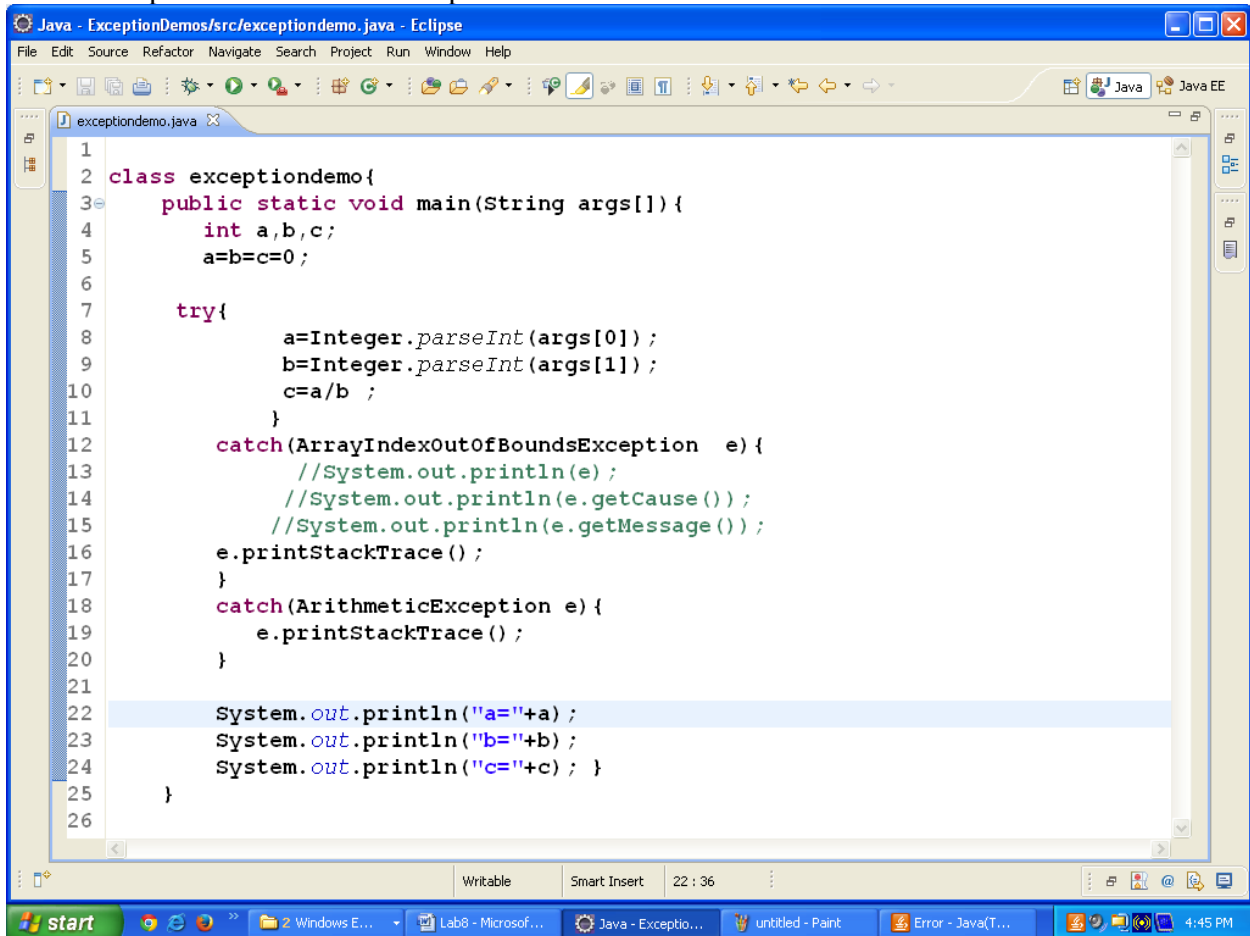


# Lab-8 Assignments

## Exception Handling

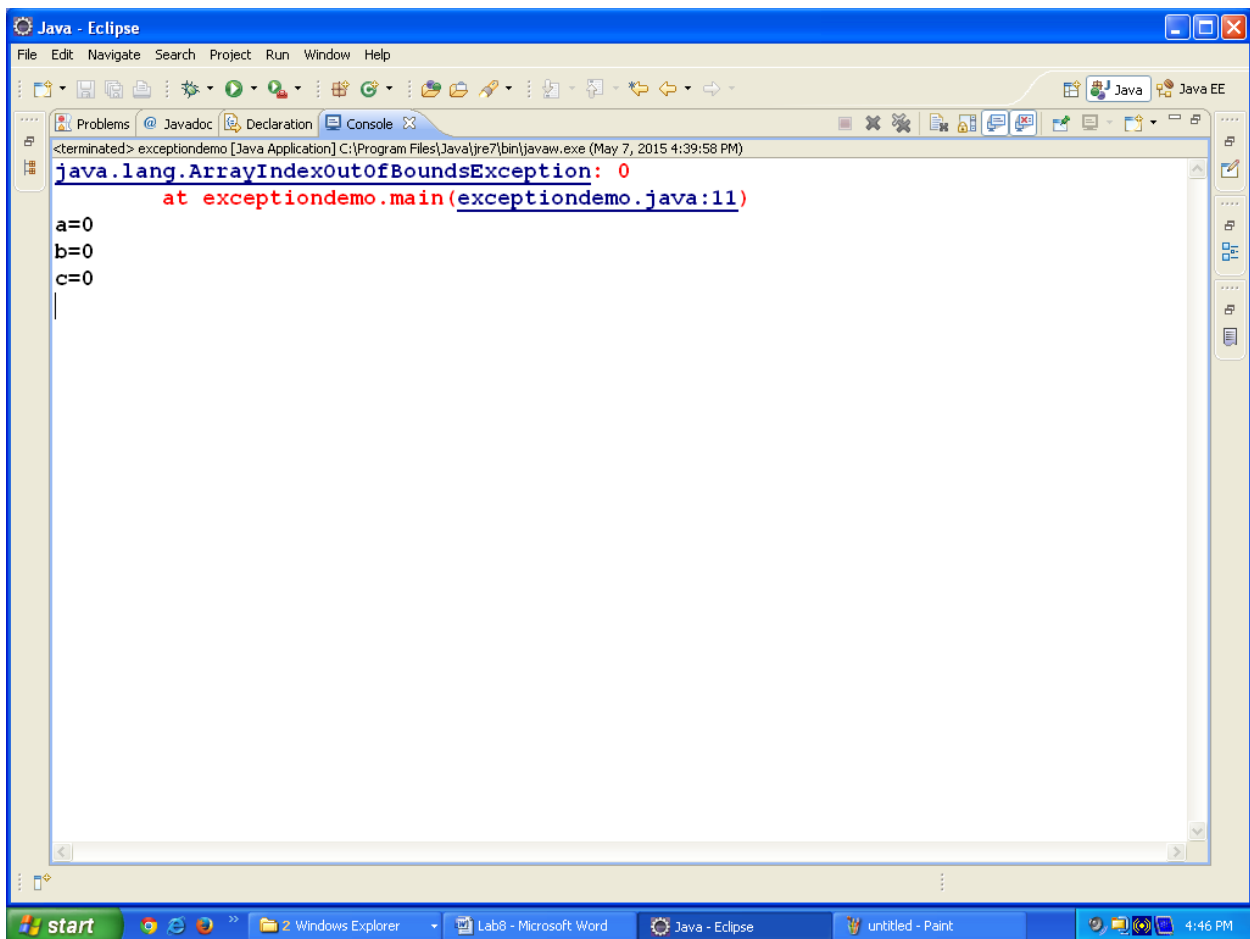
### Demo1

create exceptiondemo class in Eclipse IDE



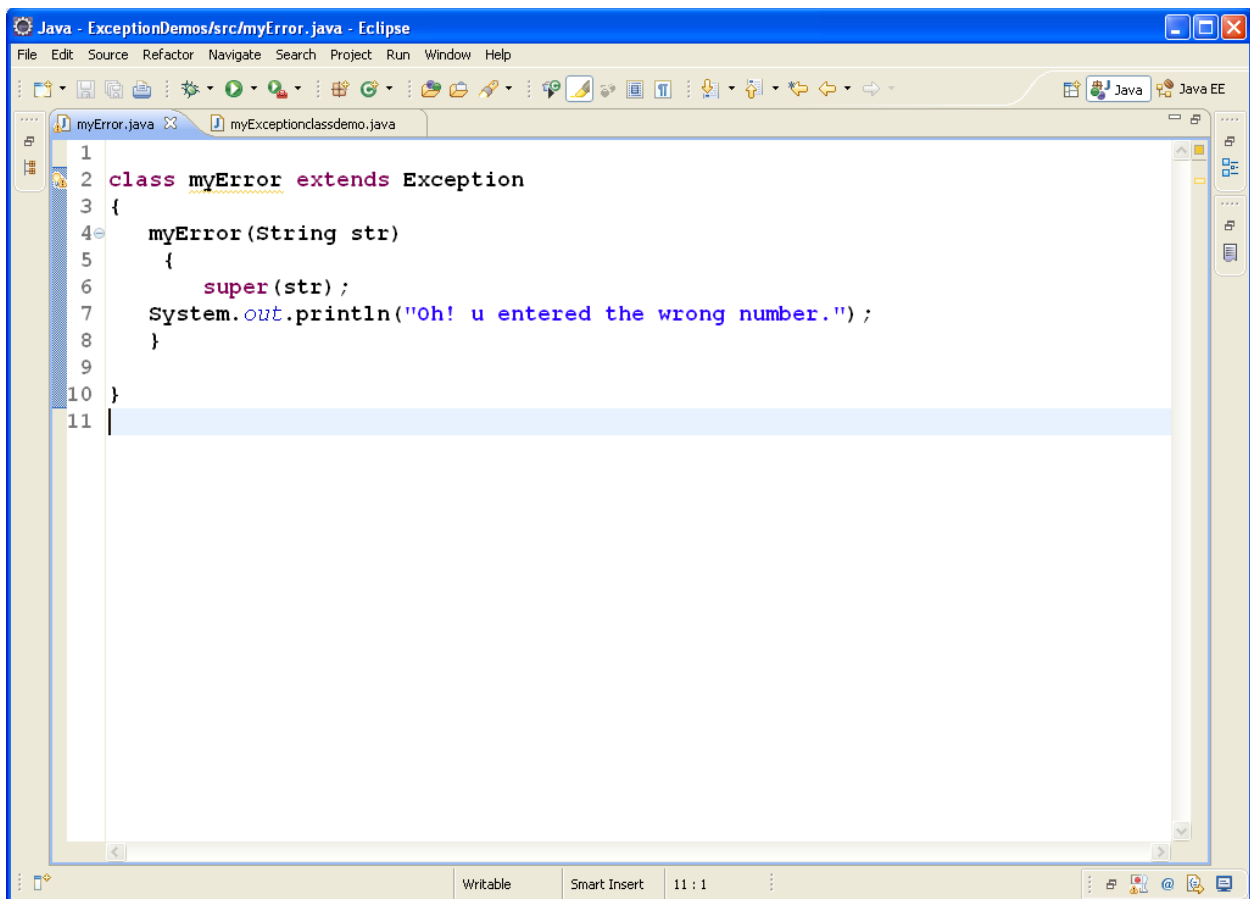
```
1
2 class exceptiondemo{
3     public static void main(String args[]){
4         int a,b,c;
5         a=b=c=0;
6
7         try{
8             a=Integer.parseInt(args[0]);
9             b=Integer.parseInt(args[1]);
10            c=a/b ;
11        }
12        catch(ArrayIndexOutOfBoundsException e){
13            //System.out.println(e);
14            //System.out.println(e.getCause());
15            //System.out.println(e.getMessage());
16            e.printStackTrace();
17        }
18        catch(ArithmeticException e){
19            e.printStackTrace();
20        }
21
22        System.out.println("a="+a);
23        System.out.println("b="+b);
24        System.out.println("c="+c);
25    }
26}
```

Run as -> Java Application



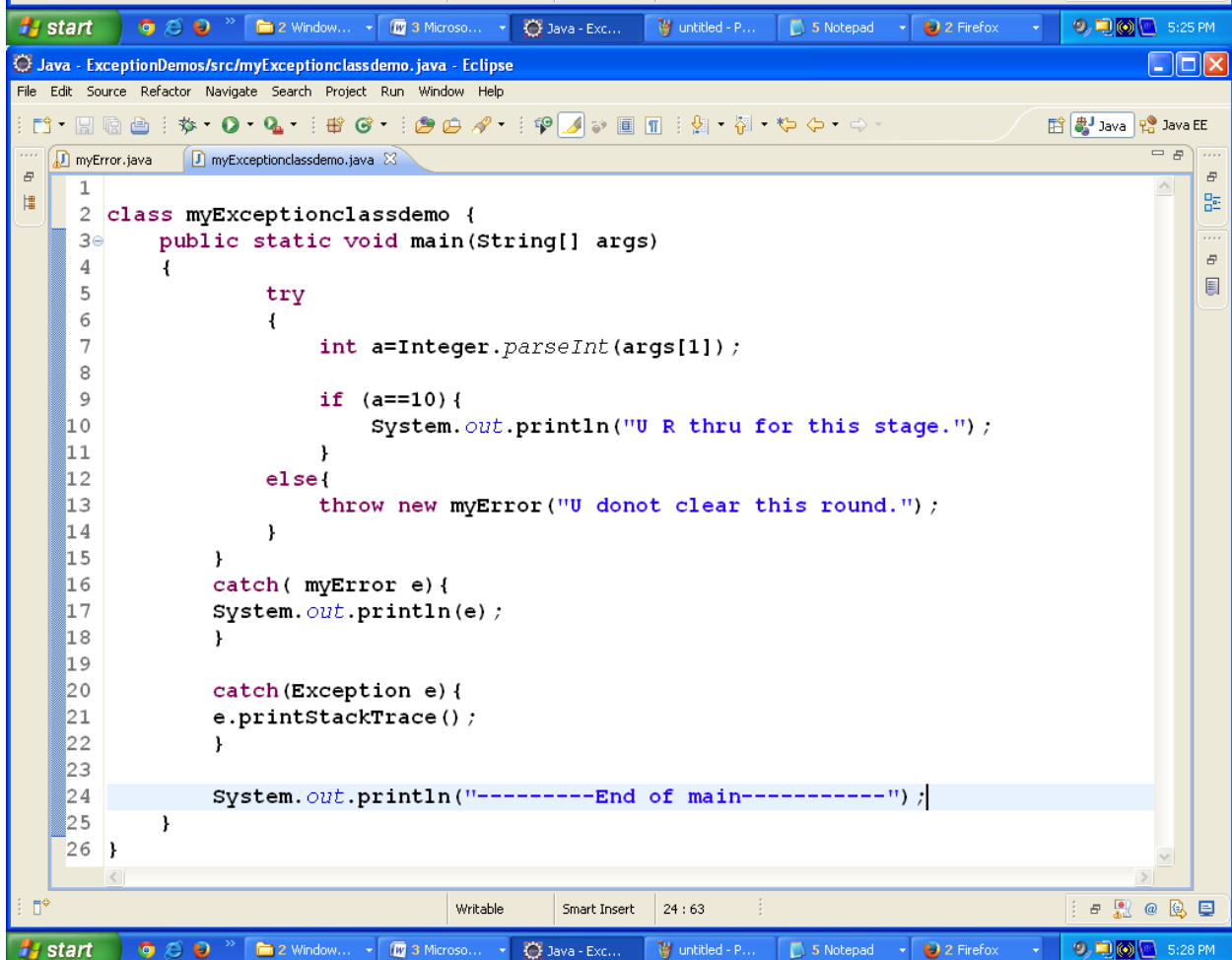
## Demo2

create userdefined exception class **myError** and **myExceptiondemoclass**



The screenshot shows the Eclipse IDE with the file `myError.java` open. The code defines a custom exception class `myError` that extends `Exception`. It has a constructor `myError(String str)` that calls `super(str)` and prints a message to the console.

```
1 class myError extends Exception
2 {
3     myError(String str)
4     {
5         super(str);
6         System.out.println("oh! u entered the wrong number.");
7     }
8 }
9
10
11
```

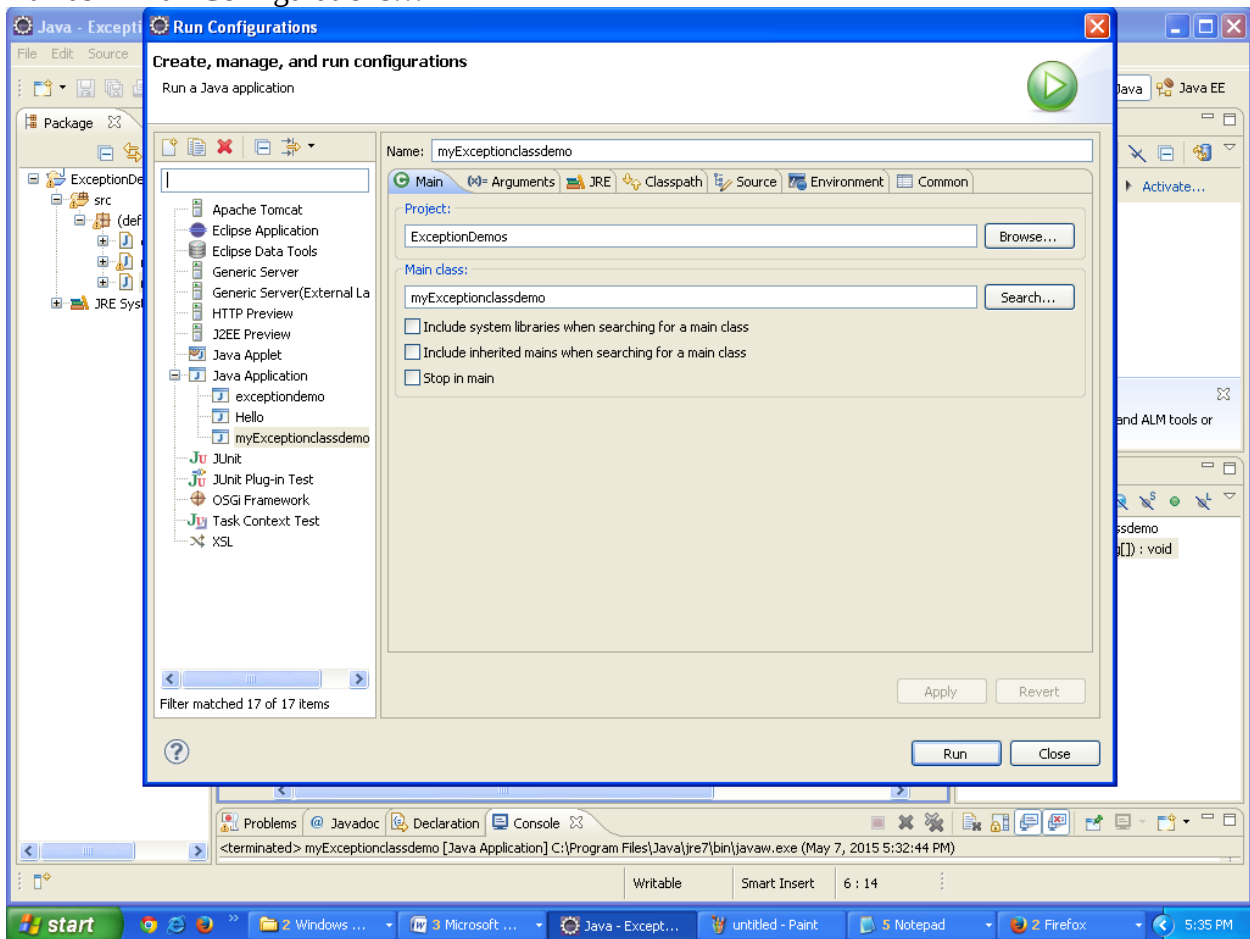


The screenshot shows the Eclipse IDE with the file `myExceptionclassdemo.java` open. The code defines a class `myExceptionclassdemo` with a `main` method. The `main` method takes command-line arguments, parses the first argument, and checks if it is equal to 10. If it is, it prints a message. Otherwise, it throws a `myError` exception. The `main` method also has two catch blocks: one for `myError` and one for `Exception`. The `main` method ends with a print statement.

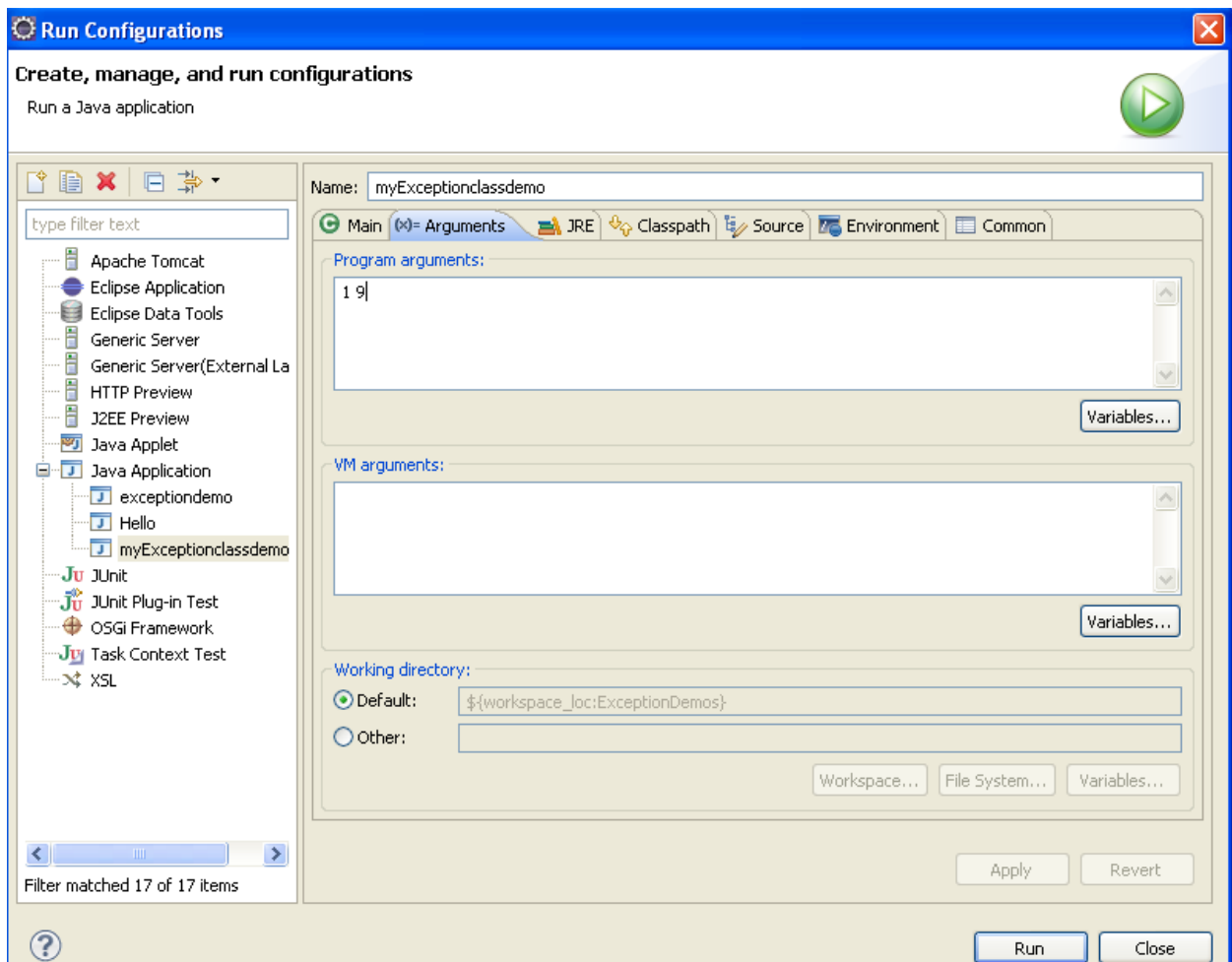
```
1 class myExceptionclassdemo {
2     public static void main(String[] args)
3     {
4         try
5         {
6             int a=Integer.parseInt(args[1]);
7
8             if (a==10){
9                 System.out.println("U R thru for this stage.");
10            }
11            else{
12                throw new myError("U donot clear this round.");
13            }
14        }
15        catch( myError e){
16            System.out.println(e);
17        }
18        catch(Exception e){
19            e.printStackTrace();
20        }
21
22        System.out.println("-----End of main-----");
23    }
24 }
25
26
```

Steps for passing CLA(command Line Arguments in eclipse)

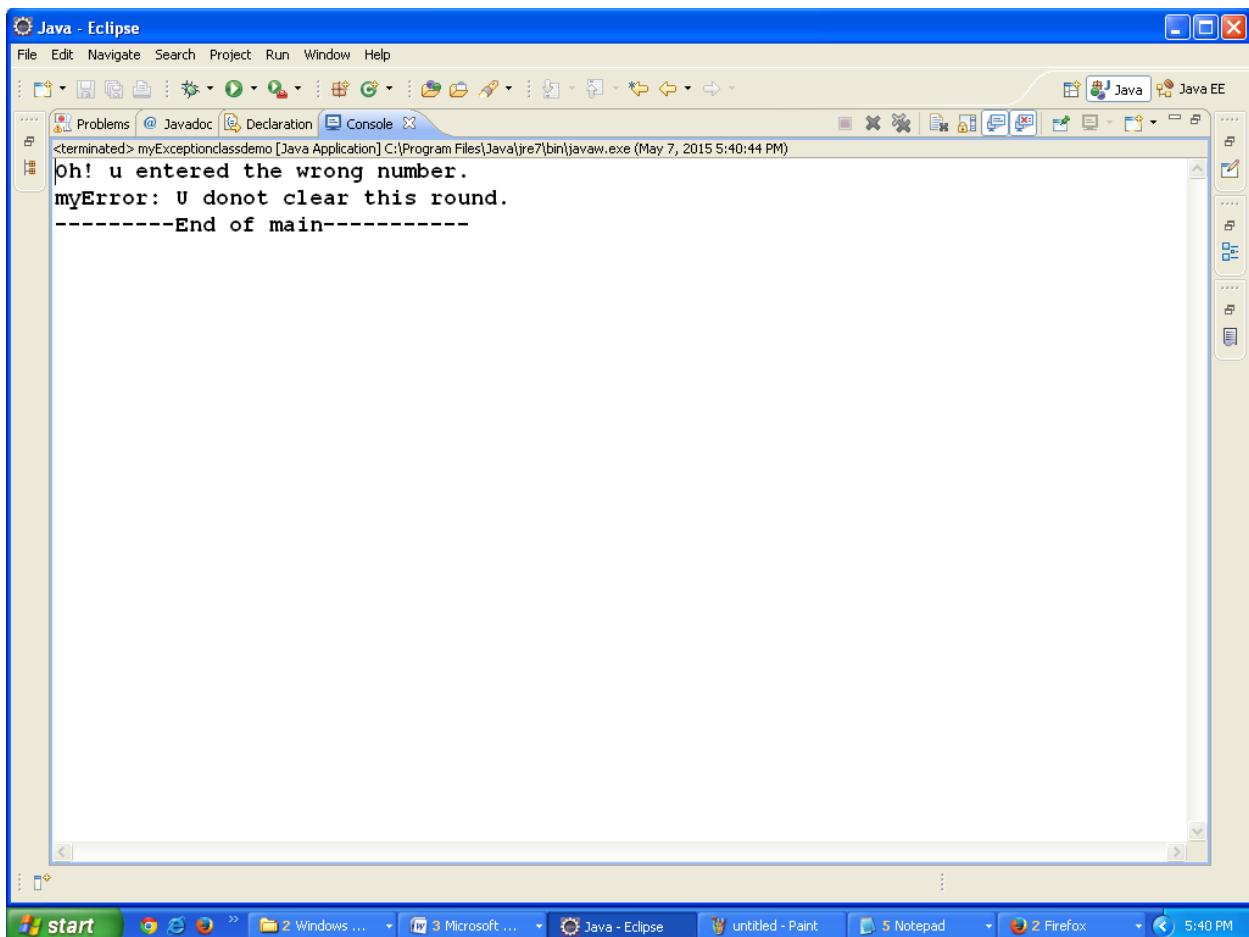
Run as -> Run Configurations...



Click on Arguments



Enter Arguments and click on Run



## Assignments To Solve:

1. Create a class called CalcAverage that has the following method:  
public double avgFirstN(int N)

This method receives an integer as a parameter and calculates the average of first N natural numbers. If N is not a natural number, throw an exception IllegalArgumentException with an appropriate message.

2. Create a class Number having the following features:

Attributes

int	first number
int	second number
result	double stores result of math operations performed on a & b

Member functions(Methods)

Number(x, y)	constructor to initialize the values of a and b
add()	stores the sum of a and b in result
sub()	stores difference of a and b in result
mul()	stores product in result
div()	stores a divided by b in result

Test to see if b is 0 and throw an appropriate exception since division by zero is undefined. Display a menu to the user to perform the above four arithmetic operations.

3. Create a user defined Exception class called **InsufficientFundsException**  
(Note: Use the existing **Account** Class (created in lab3))

Handle an exception for **Withdraw(int amt)** in Account class

**Withdraw(int amt)** should throw **InsufficientFundsException** if the amount to be withdrawn is greater than the balance.

Accept the values from user for creating Account object and amt to withdraw

4. Create a userdefined Exception classes **IncorrectAgeException** & **IncorrectNationalityException**

Write a class **Voter** with constructors and methods **toString()**, **check(String name, int age)**

The check (...) should check for Nationality and age for voting and throw appropriate Exception

Accept the Name, Nationality & age from user