

Debugging in NetBeans - Introduction

What is Debugging?

When you write programs, things don't always work correctly. That is, in fact, one of the reasons why we need to test our classes - we want to make sure they work correctly. However, when things don't work correctly, it can be difficult to figure out exactly what the problem is so that we can fix it. NetBeans, like most other programming environments, provides a particular tool for this called a **debugger**.

Debugging can be defined as the process used for examining the code for errors. Debugging is carried out by setting breakpoints in code and then using debugger to run it. We can execute our code one line at a time and examine the state of our application in order to discover any problems.

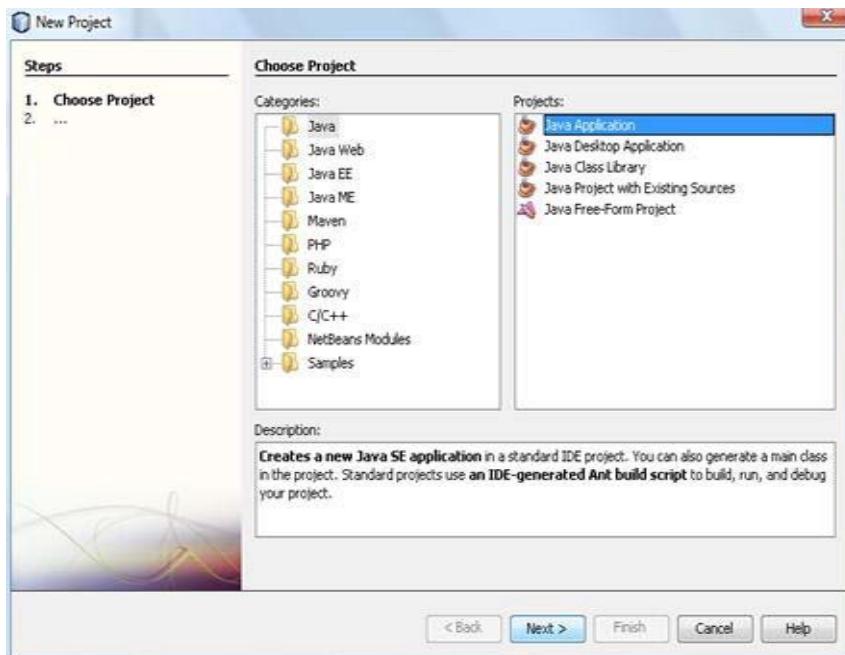
When we start a debugging session in NetBeans, all of the relevant debugger windows appear automatically at the bottom of our screen. We can debug an entire project, any executable class, and any JUnit tests.

Practice Example:

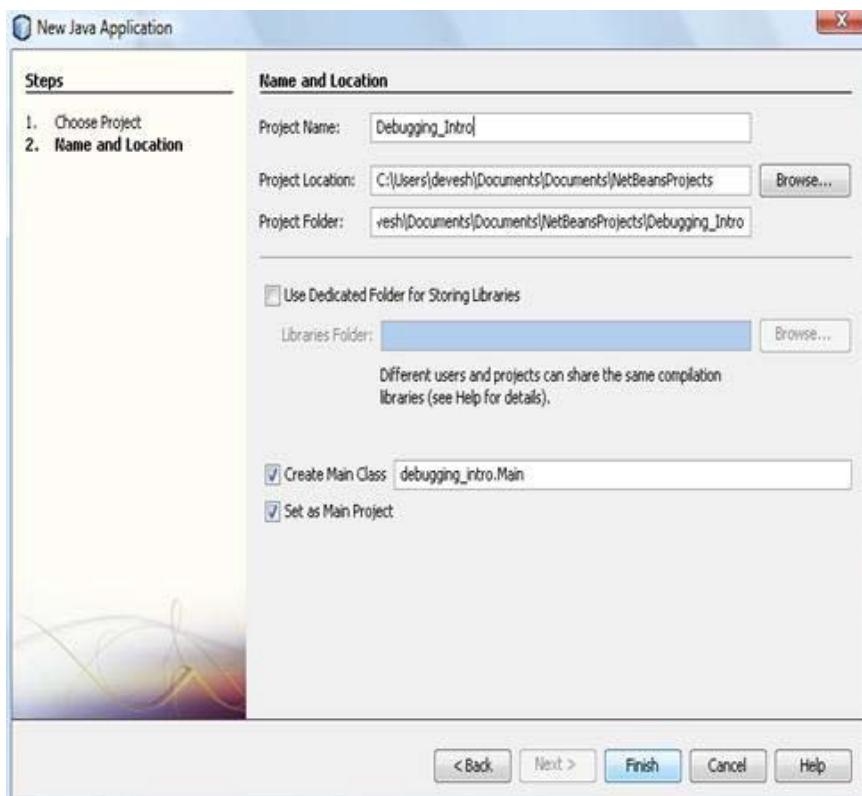
Make sure that NetBeans 6.7 IDE is open on your desktop

Go to **File** on the left top side of the menu bar and select **New Project**.

In the pop menu that appears, select **Java** under categories and **Java Application** under Projects. Click **Next >** button.



Name the project as **Debugging_Intro** and select appropriate project location.



Click on **Finish** button.

Note that a project with the name **Debugging_Intro** is being created on the left pane under projects.

Replace the main method with the following method in the main class.

```
public static int main(String[] args) {  
    // TODO code application logic here  
  
    Scanner sc;  
  
    System.out.println("Enter any month in number (Ex: 1 for January, etc)");  
    sc = new Scanner(System.in);  
    float month = Integer.parseInt(sc.next());  
  
    if (month / 2 == 0) {  
        System.out.println("The month is Even");  
    } else {  
        System.out.println("The month is Odd");  
    }  
  
    // switch statement to return the name of the month  
    switch (month) {  
        case 1:  
            System.out.println("January");  
            break;  
        case 2:  
            System.out.println("February");  
            break;  
        case 3:  
            System.out.println("March");  
            break;  
        case 4:  
            System.out.println("April");  
            break;
```

```

case 5:
System.out.println("May");
break;
case 6:
System.out.println("June");
break;
case 7:
System.out.println("July");
break;
case 8:
System.out.println("August");
case 9:
System.out.println("September");
break;
case 10:
System.out.println("October");
break;
case 11:
System.out.println("November");
break;
case 12:
System.out.println("December");
break;
default: System.out.println("Invalid
month."); break;
}

//while loop to calculate time to new year
int j = 0;
int i = month;
while (i != 0) {
i += 1;
j++;
System.out.println("Calculating time to New Year...");
}

System.out.println("Total months left for New Year:" + j);
}

```

Note: Remember that you can format your code by **right clicking** anywhere inside the class and selecting **Format**. It will align your code properly.



Notice that there are few **compilation errors** in the program.

Note: The compiler is always running behind the scene. Thus, there is no "Compile" option in this environment to choose from.

The very **1st compilation error** is in the **main** method.

Change **int** to **void** in the **main** method to remove the compilation error. The **main** method is always **void** as it **doesn't return any value**.

Note: The main method is always:

public: so that it can be accessed outside the class.

static: so that it can be invoked without creating any objects.

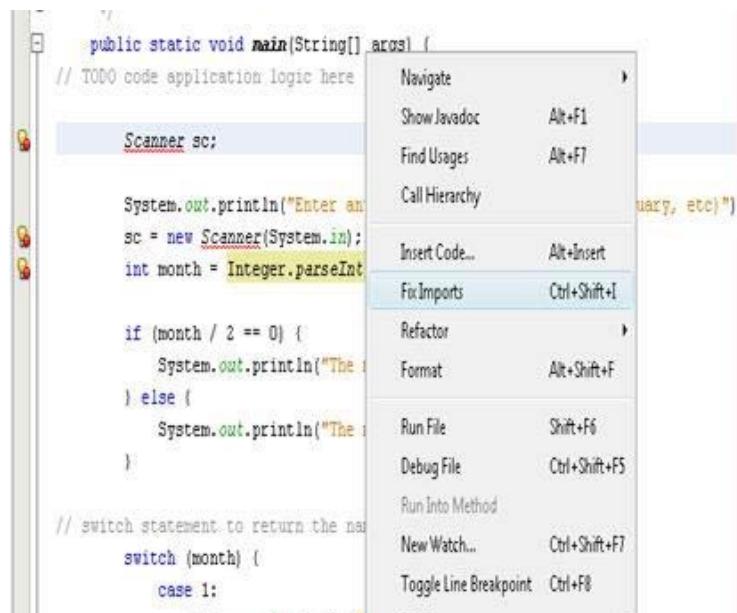
void: as it doesn't return any value.

The **2nd compilation error** is in the declaration of **switch statement**. This error is because of the fact that switch statement does not work with float.

At the start of the main method, change variable **month** to the data type **int** from float.

Note: A **switch** works with the **byte, short, char, and int** primitive data types.

To remove other errors, right click in the source editor window and select **Fix Imports**.



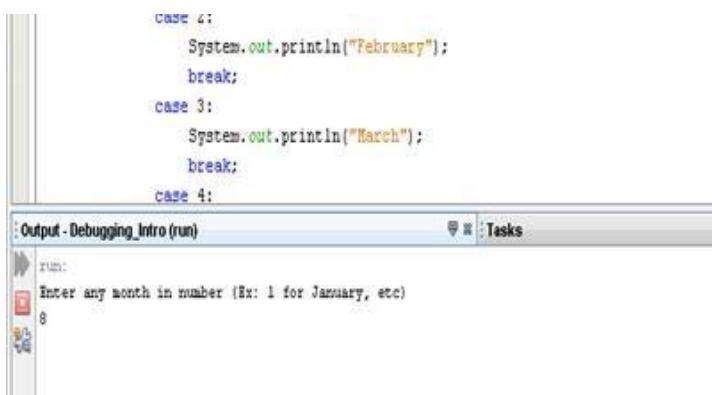
Make sure that **java.util.Scanner** appears under import statements in the pop up box. Click **OK**.



Note that now there are no compilation errors in the program.

Save the class and click on the **Run**  icon on the top of the menu bar or by right clicking on the project and select **Run**.

When the program prompts for the input, enter **8** in the Output pane at the bottom of the IDE and hit **Enter**.



```
case 2:  
    System.out.println("February");  
    break;  
case 3:  
    System.out.println("March");  
    break;  
case 4:  
  
Output - Debugging_ Intro (run) Tasks  
run:  
Enter any month in number (Ex: 1 for January, etc)  
8
```

Note that despite the entered month as even (August), the output being displayed is **The month is Odd** and **August September**. And then the program enters into an **infinite loop**.

To exit from the infinite loop, click on the close button  at the bottom-most margin of the IDE.



```
Calculating time to New Year...  
Calculating time to New Year...  
Calculating time to New Year...  
Calculating time to New Year...
```

We're now going to track down the cause of the problem using the **debugger**. Do not solve the problem just by looking at the code. The point of this exercise is to learn how to use the debugger to solve more difficult problems than this one.

We want to step through the program one step at a time, in order to figure out what is wrong with the code. To do this, we can first set a **breakpoint** in the code. A **breakpoint** is simply a flag that tells NetBeans "Stop here if you're debugging."

A **break point** is a debugging mechanism which tells the program that when you reach this point you want to stop the program. To set a **break point** you go to the line of code that you wish to set it on and then click on the grey border on the left. This will put a **breakpoint** on this line.

Let's trace the program to see why these errors occurs by placing a breakpoints on the **if** statement, **switch** statement and **while** loop. You can do this by clicking on the left grey stripe of the editor window. (Click again to remove the break point.)

```

public static void main(String[] args) {
    // TODO code application logic here

    Scanner sc;

    System.out.println("Enter any month in number (Ex: 1 for January, etc)");
    sc = new Scanner(System.in);
    int month = Integer.parseInt(sc.nextInt());

    if (month / 2 == 0) {
        System.out.println("The month is Even");
    } else {
        System.out.println("The month is Odd");
    }

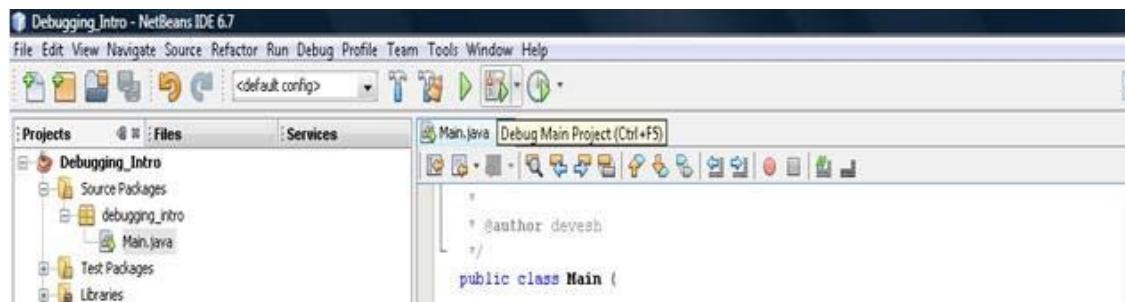
    // switch statement to return the name of the month
    switch (month) {
        case 1:
            System.out.println("January");
            break;
        case 2:
            System.out.println("February");
            break;

        //while loop to calculate time to new year
        int j = 0;
        int i = month;
        while (i != 0) {
            i += 1;
            j++;
            System.out.println("Calculating time to New Year...");
        }

        System.out.println("Total months left for New Year:" + j);
    }
}

```

Click on the **Debug Main Project** icon on the top of the menu bar.



Once again, enter **8** in the Output pane at the bottom of the IDE and hit **Enter**.

To walk through the program step by step, you'll use the **Step Over** icon in the toolbar in the Debug pane of this window, which is located on the top of the IDE.

```

    System.out.println("Enter any month in number (Ex: 1 for January, etc)");
    sc = new Scanner(System.in);
    int month = Integer.parseInt(sc.next());

    if (month / 2 == 0) {
        System.out.println("The month is Even");
    } else {
        System.out.println("The month is Odd");
    }
}

```

Click on **Step Over** and continue with the clicking of Step Over and you will find that:

- a. instead of if statement in the main method, else statement is being executed.
- b. along with case 8 in switch statement case 9 is also being executed.
- c. the program never exits the while loop (value of i keeps increasing!) and therefore this loop is being executed infinite times. In fact, click on the **Variables** tab at the bottom of the IDE and you can observe the values of i goes from 8, to 9, to 10, to 11, etc., not going towards zero at all! This is a classic infinite loop! This process of variable watching allows you to observe the behavior of your program easily.

Name	Type	Value
<Enter new watch>		
+ Static		
+ args	String[]	#44(length=0)
+ month	int	8
+ j	int	0
+ i	int	8

As we have now figured out the source of problem, stop the debugging by clicking on **Finish** icon on the top of the IDE.



We found at the time of debugging that the if condition is not properly written.

To fix this error, replace the '/' operator in the if statement with '%' operator. The **if** statement should look like this:

```
if (month % 2 == 0) {  
    System.out.println("The month is Even");  
} else {  
    System.out.println("The month is Odd");  
}
```

Note: We know that '/' is the division operator. On the other hand, the '%' is the remainder operator, which divides one operand by another and returns the remainder as its result. For example, if 8 is divided by 2, the '/' operator will return 4 and the '%' operator will return 0.

While debugging we noticed that along with case 8 in switch statement, case 9 is also being executed. This is because of the fact that there is no **break;** statement in case 8 and therefore soon after execution of case 8, the control shifts to case 9.

To fix this error, include the **break;** in **case 8** of the switch statement.

Lastly, we have also discovered while debugging that the **while** loop executes infinitely.

Look closely and you will find that as we need to calculate months remaining for new year, the **i != 0** doesn't make any sense. It must rather be **i != 12**.

Therefore to get rid of this error, replace **i != 0** with **i != 12** in the **while** loop.

Save your project and Run the project again by clicking on the **Run** icon on the top of the menu bar or by right clicking on the project and select **Run**.

Enter any month of your choice and now you will get the correct output without any more errors! :)

The purpose of this tutorial was to make you familiar with debugging by using a simple JAVA program! It is expected that you learn how to use the debugger to solve more difficult problems than this one !!