Name : Sakshi Mohan Kamble

PRN : 2020BTEIT00205

Course : Agile software tools and practices.

**Assignment 1**

**Title:** **Debug the a code using eclipse IDE**

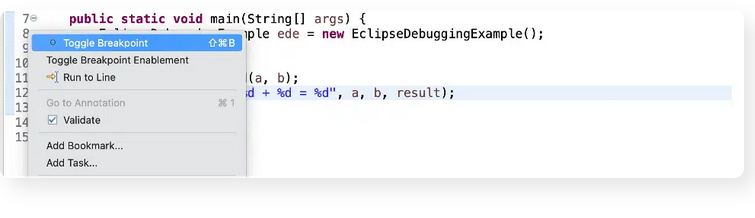
**Aim :** Debug the java code using eclipse IDE

**Methodology:**

## Step One: Set Breakpoints

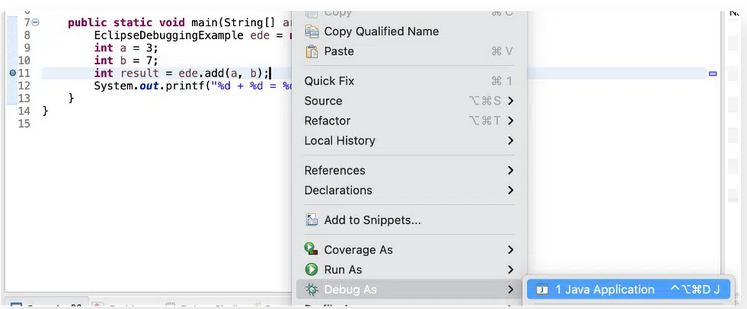
A breakpoint is a point in code where the program execution pauses during debugging. This allows the programmer to inspect code and the flow of execution at the defined breakpoint.

To define a breakpoint, either double click on the left margin in the editor or right click and select Toggle Breakpoint:



## Step Two: Start the Program in Debug Mode

To debug the application, either right-click on the file in Package Explorer or within the Java editor and select Debug As -> Java Application:



## Step Three: Add Variables to Expressions for Examination

## Any custom Java expressions (including variables) and their values can be inspected in the Expressions view. For example, to track the value of the a, b and the result variables, they can be added to the Expressions view by right-clicking on each and then clicking Watch:

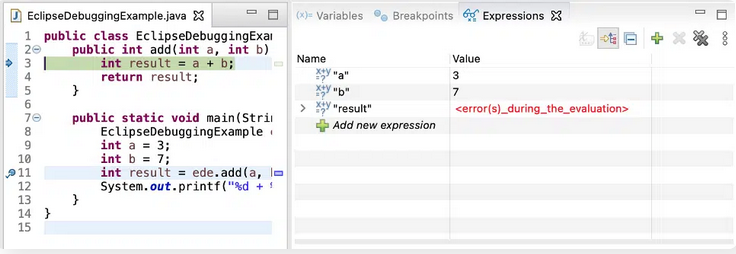
## 

## Step Four: Check the Variables Values in Expressions

## 

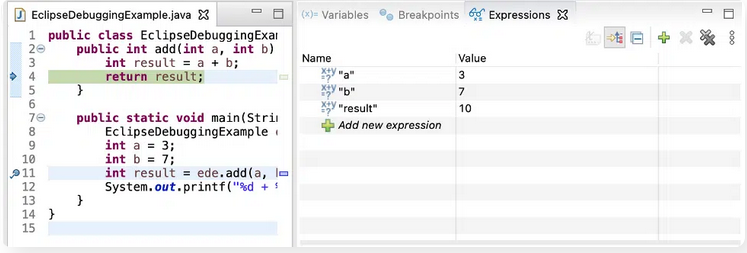
## Step Five: Step Into the Function

## Press F5 or the Step Into icon to enter into the add method:



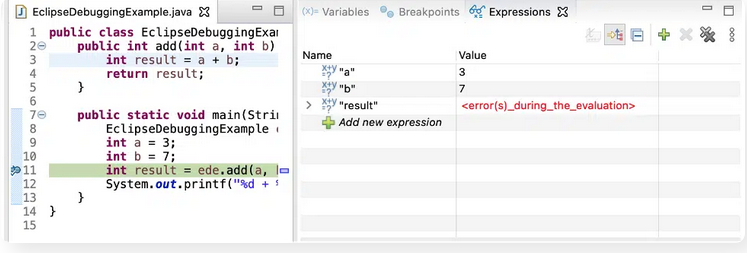
## Step Six: Step Over

Press F6 or the Step Over icon to execute the current line of code and go to the next one:

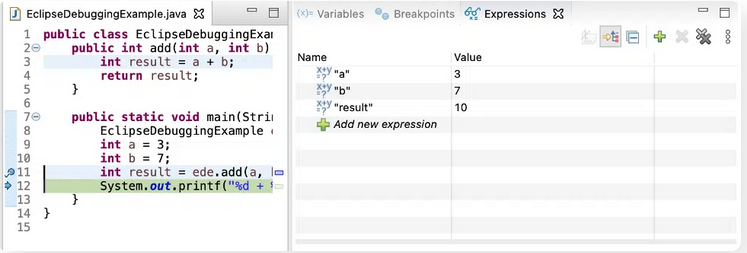


## Step Seven: Check the Return Value from Function

Press Step Over again to return to the main() method:



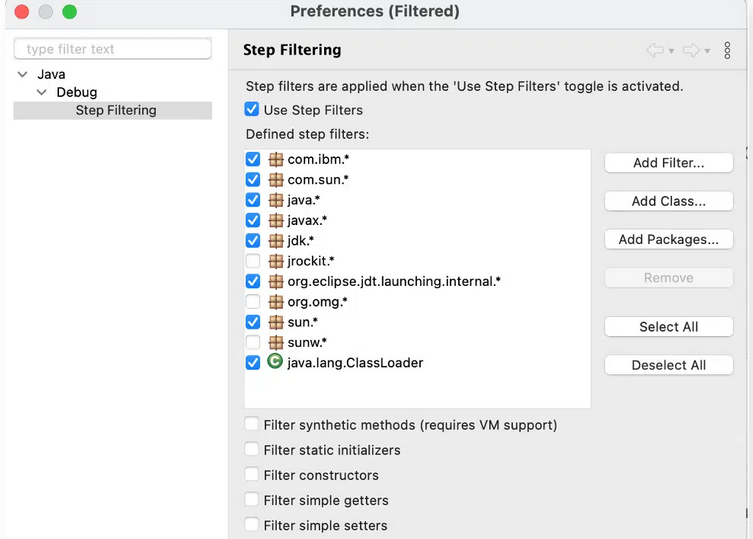
The debugger returns to stop on the same line where it left previously in Step 5. Press Step Over again to check the return value from the add() method:



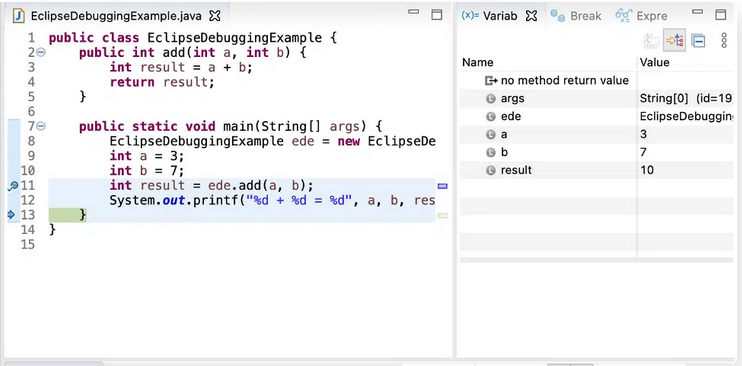
## Step Eight: Set Debug Filter

## 

In the popup window, check the Use Step Filters box. Enable the packages that you want to filter out and press Apply and Close:



Now when you press Step Into, it will no longer go to the Integer.class code:



## Step Nine: Print The Result

## 

**Code :**

public class EclipseDebuggingExample {

public int add(int a, int b) {

int result = a + b;

return result;

}

public static void main(String[] args) {

EclipseDebuggingExample ede = new EclipseDebuggingExample();

int a = 3;

int b = 7;

int result = ede.add(a, b);

System.out.printf("%d + %d = %d", a, b, result);

}

}