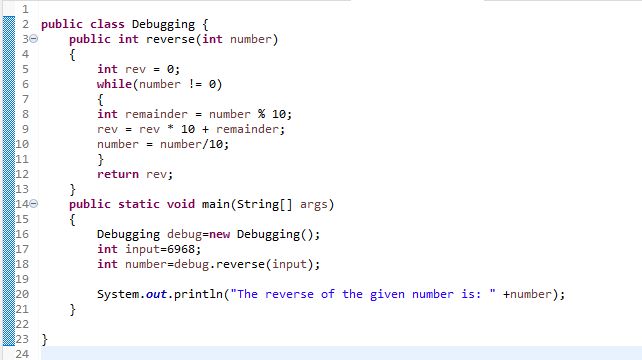
**Debugging Your Code in Eclipse**

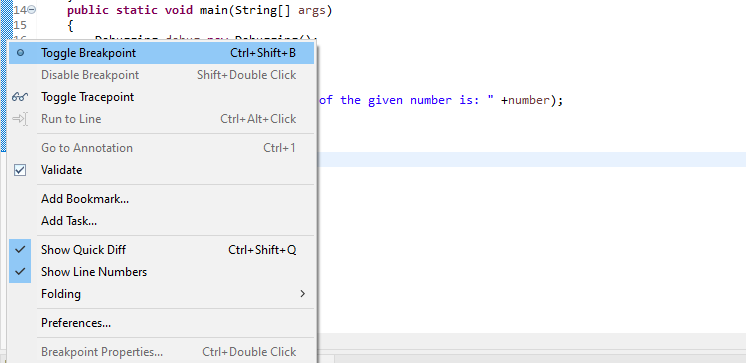
* + Debugging is the process of identifying and fixing any issues in the source code of a program.
  + Modern IDEs like Eclipse provide debugging tools that make it easier for developers to walk through their code interactively and inspect it to spot and resolve any issues.

**Example:**

****

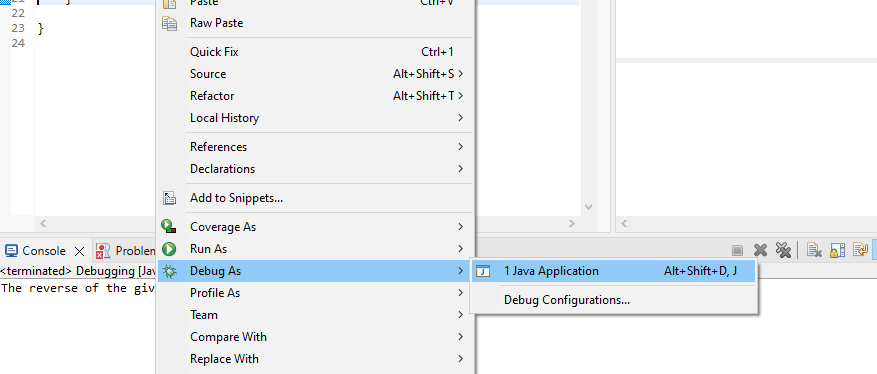
## **Step 1 : Set Breakpoints**

* + 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:

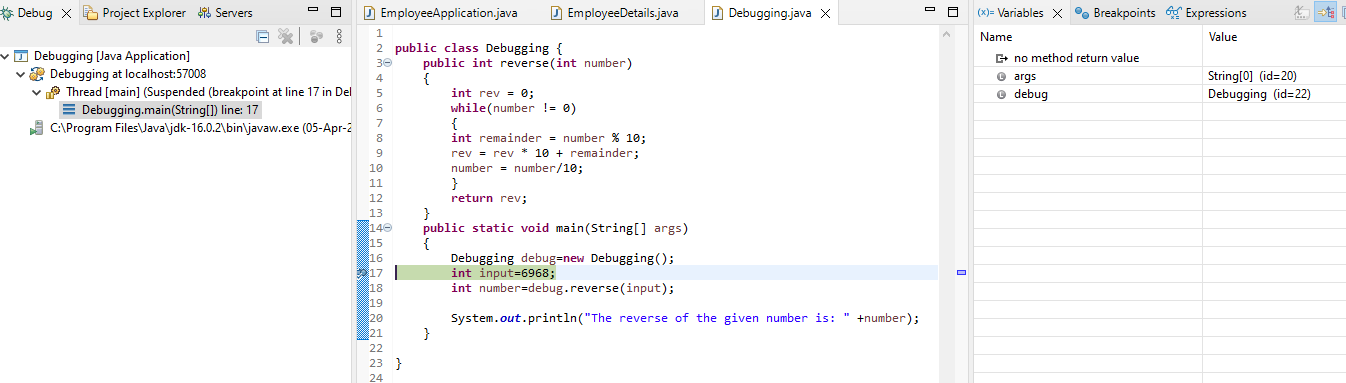


## **Step 2 : Start the Program in Debug Mode**

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



If you click Switch, Eclipse opens this perspective, which has a group of views that help with the debugging process:



The most commonly used views are described below:

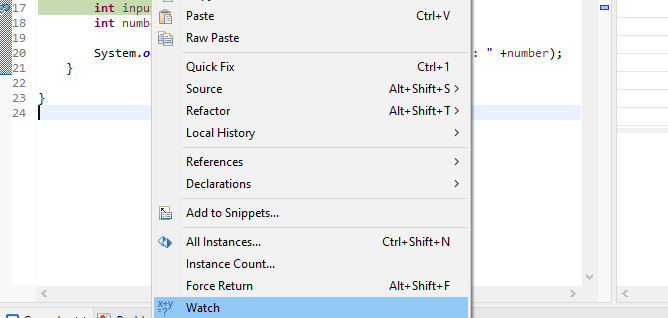
* ***Debug***: Displays the call stack which helps in determining the flow of execution of the program until the breakpoint is reached
* ***Variables***: Displays fields and defined variables in the current stack
* ***Breakpoints***: Shows a list of all breakpoints in the code and enabling/disabling breakpoints
* ***Expressions***: Allows defining custom Java expressions to inspect their values

Eclipse also provides several buttons in the toolbar for controlling the flow of execution of the program:



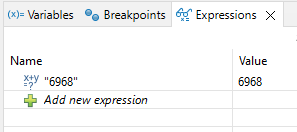
## **Step 3 : Add Variables to Expressions for Examination**

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



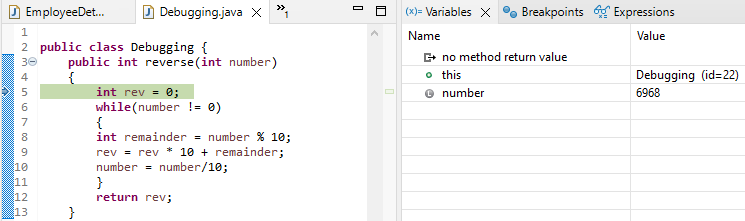
## **Step 4 : Check the Variables Values in Expressions**

The variables are now visible in the *Expressions* view:



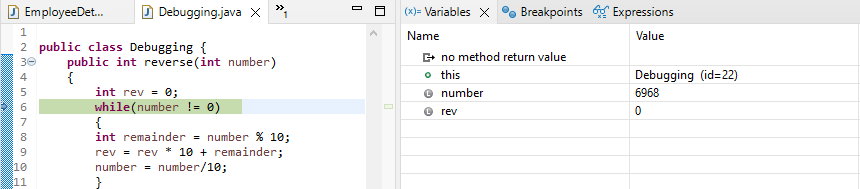
## **Step 5 : Step Into the Function**

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



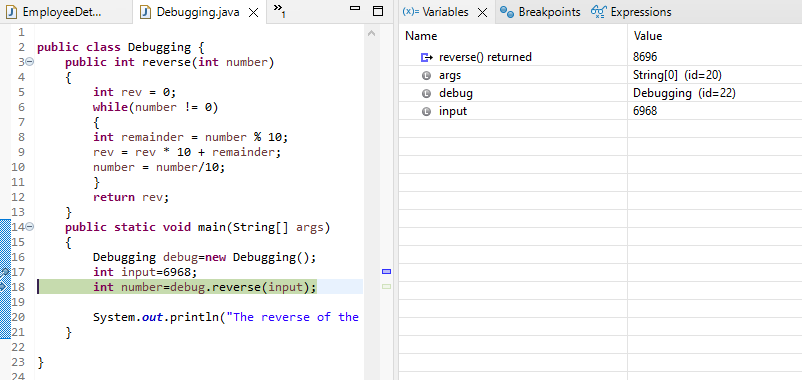
## **Step 6 : Step Over**

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

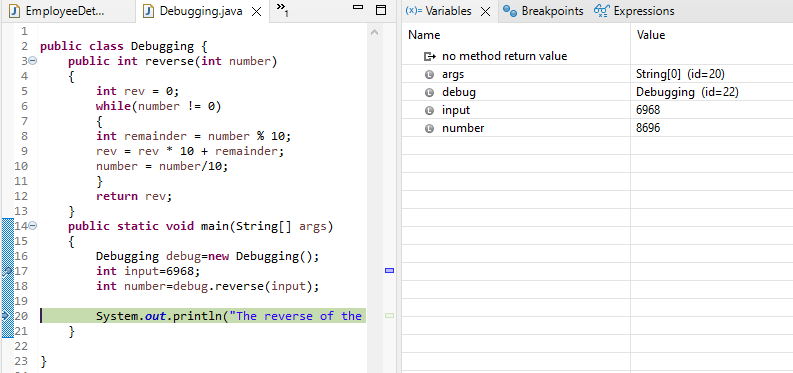


## **Step 7 : Check the Return Value from Function**

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

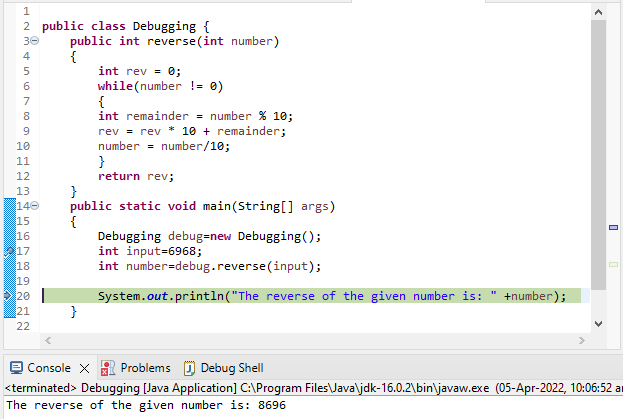


The debugger returns to stop on the same line where it left previously in Step 5. Press Step Over again to check the return value from the reverse() method:



## **Step 8 : Print The Result**

At this point, line 20 executes and prints the result to the Console successfully:



## **Step 9 : Stop in Main**

The debugger can now be stopped by pressing the *Terminate* icon in the toolbar.



The debugger will now be disconnected:

