**1.Bug**

**2.Debugging**

**3. Debugger**

**4.Debugging steps**

**5. Debugging In Eclipse IDE.**

**5.1.BreakPoint**

**5.2. Debugging Operation.**

**5.3. Debugging the web Application.**

**1.BUG:-** Bug is coding error in computer program.

Bur is flaw,error in designing.

**2.DEBUGGING:-** Debugging is process of finding and fixing(removing) **logical errors** or bugs in source code of any software. When software does not work as expected or when software will not give expected result.

By logical error , we don’t get CE,RE . But we don’t get the expected result from the software /program.

Advantages:-

* After joining in the company and After assigning to existing project, programmer has to perform the debugging process to know execution flow in the project.
* By identifying errors at an early stage can solve a lot of time to developer.
* During debugging, we can watch variable current value just by placing the cursor on variable. We can also modify variable current value during debugging.

**3.DEBUGGER:-** The tool or program or software that can be used for debugging is called **debugger/Debugger** tool.

There are two types of Debugger tools:

**3.1.JDB:-**

It is supplied by “oracle” along with jdk. It is installed along with jdk.

It is available in <java\_home>/bin as “jdb.exe”.

It is CUI tool and not so popular.

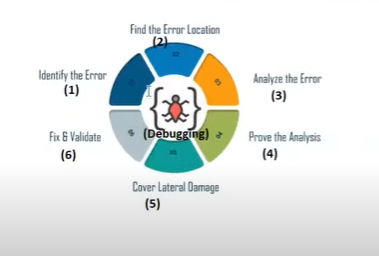
**3.2 IDE supplied Debugger:-**

IDEs(eclipse, net beans,…etc) supplies debugger tools.

They are GUI tools .

Industry recommended them.

**4. Debugging Stages/steps:-**

****

4.1. Identify Error

4.2. Identify the Error Location:- we need to identify exact location of error in code where an error occurs. Identifying an exact location that leads error can help to fix problem faster.

4.3. Analyze the Error:- we should analyze the error to understand the problem.

4.4. prove the analysis:- we need to write the test cases. By test cases , we should prove the our error analysis.

4.5. Cover lateral Damage:- when modify code to resolve bug, the current bug will be fixed but new issue may occurs. Such problems will be identified and such problems are also to be fixed.

4.6. fix & validate.

**5. Debugging In Eclipse IDE**

**1.Break point:-** Breakpoint is point in the program/application/software from there debugging will start. Till breakpoint , code executes normally. From the break point ,execution control will be handovered to programmer.

**Types of Break point:**

1. Method Break point:-

Symbol is  . This break point is set before the method header statement.

1. Line Break point:-

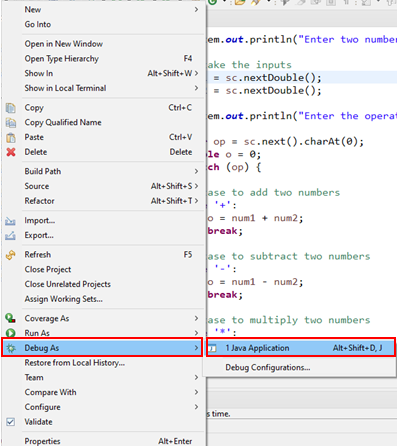
Symbol is . This type of break point is set before the line which is inside the method body.

There are two ways to set/unset the break point in code.

1. Place the cursor on a line in code and press the ctrl+shift+b to set and to unset the break point.
2. Double click on left margin of line on which you want to set break point.

Run the java application in “Debug mode”. There are two ways to run the java application in debugmode.

1. Right click on project and select “Debug as”.

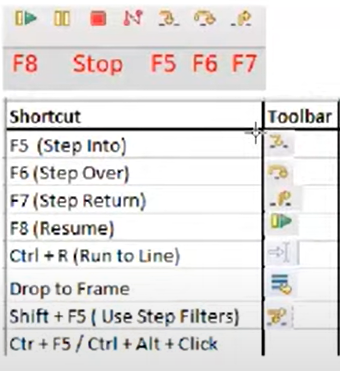


1. On right corner of eclipse editor have an icon “debug” click on it.

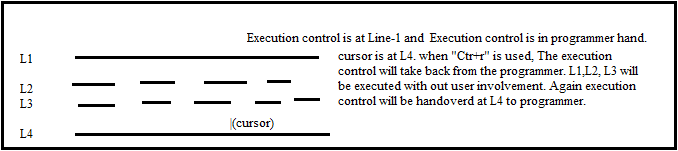


Note:-1The Eclipse IDE comes with default debugger tool. We don’t need to install debugger separetely.

**2. Debugging Operations:-**

****

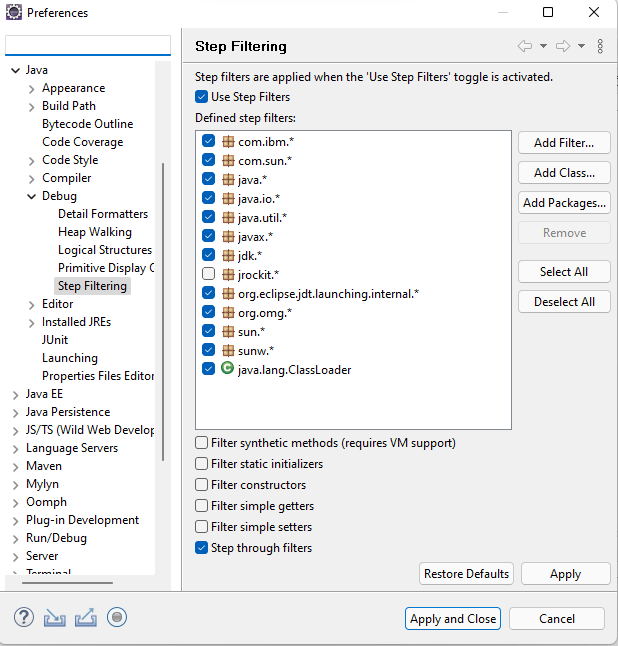
1. **Stepover :- (short cut –F6)** This operation processes the current line and proceeds to the next line. The execution control of user-defined method or pre-defined method body is not given to programmer.
2. **Stepinto:- (short cut-F5)** This operation will bring programmer into method body and This operation will give user-define method body or predefined method body execution control to programmer.
3. **StepReturn: - (short cut-F7)** when the method body is debugging, we wanted to debug remaining method body at once and we wanted to go back to caller method. In this case “Step Return” operation is usefull. This operation bring execution control from current method to caller method.
4. **Resume:-** **(shortcut-F8)** This operation makes jumping the execution control from one break point to next break point in execution flow.
5. **DropToFrame:-** This operation will bring execution control to beginning of method irrespective of whether break point is there or not.
6. **Run To Line:**- (short cut- ctl+R)



1. **Step to Selection:**
2. **Use Step Filters :- (short cut :shift+f5)** By using this shorcut, we do not make participating any pre-defined packages in debugging.

**Process:**

**Window 🡪 preferences🡪 java🡪 debug🡪 step filtering**

****

**Add the packages to step filters and select the check box.**

**Press on apply and close.**

**Note:- java.io.\* was added to step filtering.**

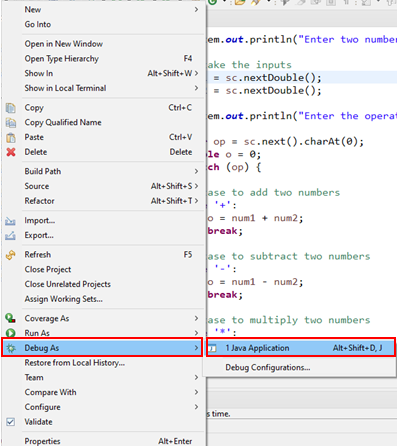
**At S.o.println(-) method, though we press the F5, we will not go into the body of println(-) method.**

**3.Debugging the web application:**

**========================**

Run the java application in “Debug mode”. There are two ways to run the java application in debugmode.

1. Right click on project and select “Debug as”.



1. On right corner of eclipse editor have an icon “debug” click on it.



**Next we can do all debugging operatios on web project code.**