Java Development with Eclipse
Lab 1 The resource perspective

1. Creating the Project, package and java classes

Create a Project, File>New>Project,

1. Select Project type 'Java'. Enter Next.
2. Project name is 'Lab'. Enter 'Finish'.

Create a Package, File>New>Package,

1. Enter 'Lab' in 'Source Folder' text window.
2. Enter 'eclipseLab' in 'name' text window. Enter 'Finish'.

Create a Class, File>New>Class,

1. Enter 'Lab' in 'Source Folder' text window.
2. Enter 'eclipseLab' in 'Package' window.
3. Enter 'MemGrab' in 'name' window.
4. Unselect all of the 'tick boxes' in the bottom 3 boxes.
5. Enter in 'Finish'

Enter the following code in the generated 'MemGrab' class template:

```
/*
 * Created on 10-Jul-2004
 *
 * To change the template for this generated file go to
 * Window&gt;Preferences&gt;Java&gt;Code Generation&gt;Code and
Comments
 */
package eclipseLab;

/**
 * @author IBM_User
 *
 * To change the template for this generated type comment go to
 * Window&gt;Preferences&gt;Java&gt;Code Generation&gt;Code and
Comments
 */
public class MemGrab {
      public MemGrab()
      {
            stringBuffer buf= new StringBuffer(10000);
      }
}
```

Create another Class, File>New>Class,

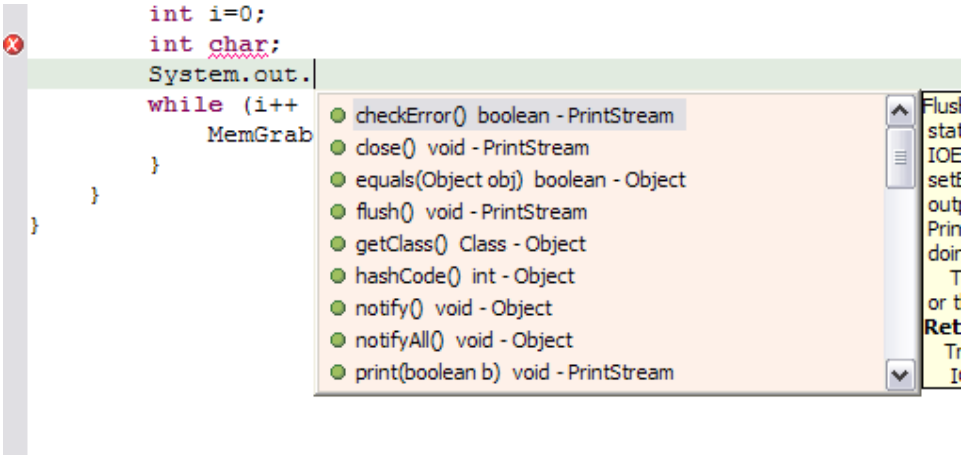
1. Enter 'Lab' in 'Source Folder' text window.
2. Enter 'eclipseLab' in 'Package' window.
3. Enter 'RunIt' in 'name' window.
4. Tick the 'public static void main(String[] args)' 'tick box'
5. Enter in 'Finish'

Enter the following code in the generated ‘RunIt’ class template:

```
/*
 * Created on 10-Jul-2004
 *
 * To change the template for this generated file go to
 * Window&gt;Preferences&gt;Java&gt;Code Generation&gt;Code and
Comments
 */
package eclipseLab;

import  EclipseLab.MemGrab;

/**
 * @author IBM_User
 *
 * To change the template for this generated type comment go to
 * Window&gt;Preferences&gt;Java&gt;Code Generation&gt;Code and
Comments
 */
public class RunIt {

      public static void main(String[] args) {
            int i=0;
            int char;
            System.out.println("Lab1");
            while (i++ < 1000) {
                  MemGrab aGrab = new MemGrab();
            }
      }
}
```

2. Code Assist

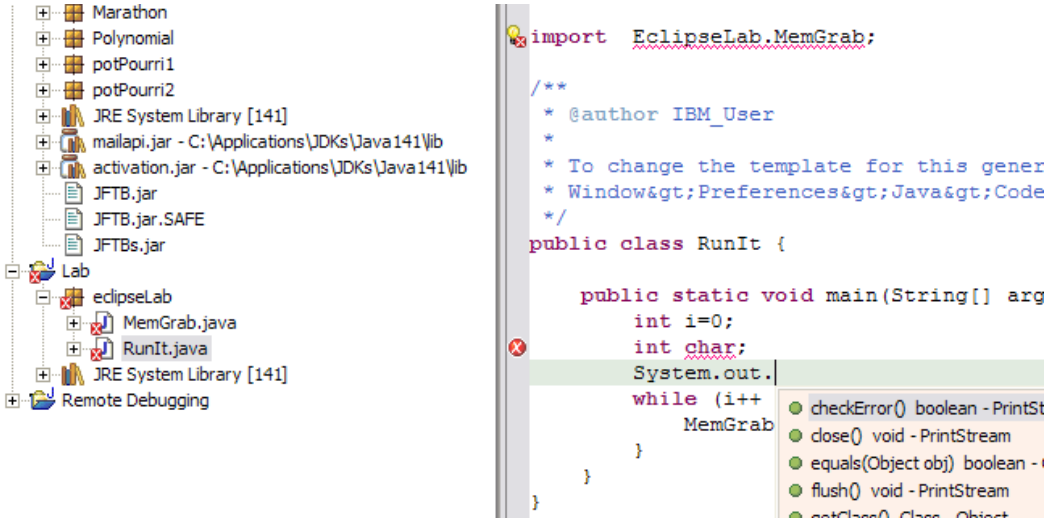
When entering ‘System.out.println’ notice how after entering ‘System.out.’ Eclipse displays a list of proposed methods.



3. Correcting the code errors

Code errors have been 'placed' in the code to demonstrate:

1. Icons associated with errors and warnings. Hint, hold the mouse cursor over the error / warning icon for more information.
2. Places where errors and warnings are identified
3. Compilation is done automatically (on file save)



The errors, and the code you need to change, is:

In RunIt.java

1. Change import EclipseLab.MemGrab to import eclipseLab.MemGrab
   The package name is 'eclipseLab' not 'EclipseLab'
2. Remove 'int char;' char is a reserve word and can't be used as an identifier.

In MemGrab.java

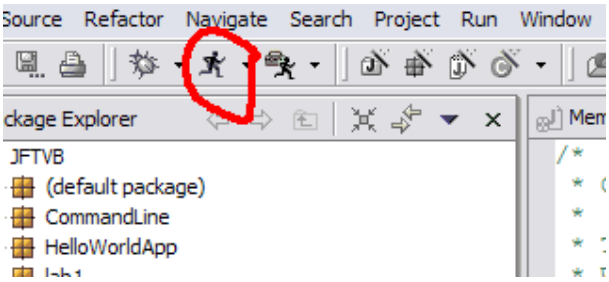
1. Change stringBuffer to StringBuffer, which is the correct class name.

You're left with 2 warnings: the local variable is never read.

4. Running the code

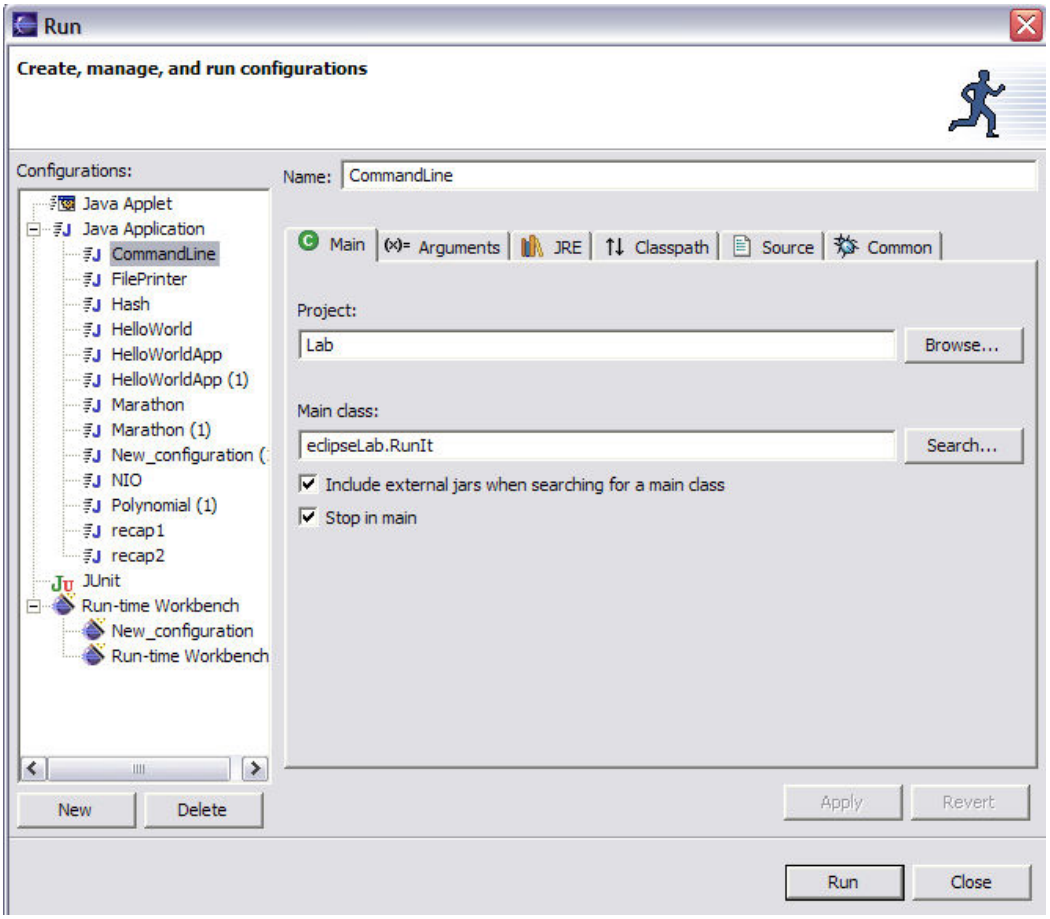
To run the application eclipseLab.RunIt() either:

1. 'Click' the run button



or 2. From the menu, Run>Run

Brings up the following panel:



Check the entries in the Project and Main class windows and ‘click’ on the ‘Run’ button.

Notice that Running the program changes the bottom left hand window from showing tasks, or the errors and warnings, to showing the console. In this example only ‘Lab2’ is displayed on the console.

5. In case you're interested, verbosegc output

Options can be passed to the JVM via the Arguments tab on the Run>Run configuration panel. Try passing '-verbosegc', as shown below, this JVM options write details of Garbage Collection activity to the console.
The example application creates a lot of large objects which results in Garbage Collection activity.