

An integrated development environment (IDE) is an application that runs on your computer and provides a text editor, program builder, program runner, and program debugger. Some IDEs are very powerful, designed for professionals, and have steep learning curves. But, the jGrasp IDE is designed for students and easier to learn. This worksheet walks you through the basics of getting started. For more information and tutorials go to <https://jgrasp.org>.

- 1) Start the jGrasp application (it should be on the Start menu in the labs).
- 2) Find the File menu and choose New > Java. This creates a blank page for you to type into..
- 3) Type the following program into the window. Each time you need to indent, push the tab key (it will turn the tab into a few spaces).

```
public class Hello {  
  
    public static void main(String[] args) {  
        String s1 = "Hello";  
        String s2 = "world!";  
        System.out.println(s1 + " " + s2);  
    }  
  
}
```

- 4) Java requires the file be named the same thing as the public class inside. So go to the File menu and Save the file with the name "Hello.java". Save it wherever you like to save files that you're working on.
- 5) Run your program! You can do this by selecting Run under the Build menu or by pressing the little red running man button. In the bottom IDE window pane you will see error messages if you mistyped something or you will see Hello world! if it worked correctly.

Next let's see how to use the debugger to watch how the program works.

- 6) Put the cursor in the line that defines s2 then right-click and choose Toggle Breakpoint. This should put a small stop sign to the left of the line. As a shortcut you can click in the grey area to the left of a line to turn on and off the breakpoint.
- 7) Start debugging your program by selecting Debug under the Build menu or clicking on the ladybug button. The left-hand IDE window pane should

show your variables. s1 should be "Hello". s2 should not be shown because the program paused just before executing the line that defines it.

- 8) In the upper-left corner of the IDE window is a row of buttons for controlling the debugger. Press the down arrow. This tells the debugger to execute the current line. Now you should see s2 has been defined and is "world!"
- 9) Great! Next let's practice everything we just did on a new example. Make a new file and type the following program.

```
import java.util.Random;

public class MaxRand {

    public static void main(String[] args) {
        Random rand = new Random();
        int max = rand.nextInt(10) + 1;
        for (int i=1; i<5; i++) {
            int next = rand.nextInt(10) + 1;
            if (next > max) {
                max = next;
            }
        }
        System.out.println(max);
    }
}
```

- 10) Study the program and see if you can figure out what it is doing. The calls to nextInt(10) evaluate to random integers in the range 0..9.
- 11) The program generates five random numbers in the range 1..10 and prints the largest of the five. Run the program with the debugger to determine what the smallest of the five numbers is.