



# The Hitchhiker's Guide to Using Java at the UCF Local Programming Contest



## Contest Conventions:

Each problem in this contest will have a filename printed at the top of the problem's specification, under the title. This filename must be used to name the file containing your solution's source code. For example, if a problem's filename is `problem`, your Java solution must be in a file named `problem.java`.

Only one class in your solution will have a `main` function. Name this class using the same filename, in all lower-case characters. For example, if a problem's filename is `problem`, then your class definition should begin:

```
class problem {
```

Your class should be in the *default package*. This means **do not** specify a package with the `package` keyword (if your editor adds a package statement to your code, you must delete it). You may use inner classes, and you may specify more than one class in your source code, but you may not submit more than one source file for a problem.

Your solution program must read the problem's input from the *standard input stream*. In Java, this is the built-in object `System.in`. Correspondingly, your program must write its output to the *standard output stream*, for which Java provides `System.out`. A few brief samples are provided below. Any other input or output could result in your solution being judged incorrect, so stick to these guidelines. If you are a C or C++ programmer, please see the corresponding version of this document.

## Example Use of the Standard Input Stream:

To read in from `System.in`, it is common to use a `BufferedReader` object. Here is template code:

```
BufferedReader stdin = null;
try
{
    stdin = new BufferedReader(new InputStreamReader(System.in));
} catch (Exception e) { }
```

You can now read in one line at a time via the `readLine` method (or use other reading methods):

```
String line1 = null;
try
{
    line1 = stdin.readLine();
} catch (Exception e) { }
```

When you reach end-of-stream, `readLine` will return `null`. You may find it useful to do something inside the `catch's {}`-block for debugging purposes. Remember to `import java.io.*;` in the top of your program.

## Example Use of the Standard Output Stream:

You should generate output using the `print`, `println` or `printf` methods of `System.out`.

```
int i;
double d;
string s;
...
System.out.println("Hello World");
System.out.printf("%d %0.3lf %s\n", i, d, s);
```