

# Doing USACO Competitions Using Eclipse

## *Do The Following The Day Before Beginning The Competition*

1. You should be familiar with the USACO rules before the day of the competition.
2. Place your hard copy resources (Battle Binder, Textbooks, et cetera) and 5-10 sheets of scrap paper, pencil/eraser and pen on your desk/table in your physical work area.
3. Pretest that your ID and password work before the day of the competition.
4. Place your electronic resources such as source, web page with important links, etc on the hard drive or CD drive. This includes ensuring that a copy of your template is functions as follows.
5. Launch Eclipse. Choose a new workspace: **File** → **Switch Workspace** → Use “**Browse...**” and “**Make New Folder**” button if necessary or, if preferred, do this in *My Computer* or a *DOS window*. Name the new workspace “**USACO**”.
6. Create a new project: **File** → **New** → **Project...** → **Java Project** → **Next** → Project Name: “**21April2006**” → Set compiler compliance to **5.0** → **Use project folder as root for sources and class files**. (This ensures that input and output data files will be easily found here.) → **Finish**.
7. **Do Not Create a Specific Package**. Use the default package by not creating a new package. This ensures that input and output data files will be automatically found when executing a program.
8. Go to Eclipse. Create a new class containing the source code of your code bank: **File** → **New** → **Class** → Name code bank class “**code**” which is saved as code.java
9. Create the input data file to test the code bank. Remember to use the extension “in”. **File** → **New** → **File** → File name: **code.in** → **Finish** → Go to class file. → Highlight the template’s data. → Copy with **[Ctrl][c]** → Return to data file. → Paste with **[Ctrl][v]** → Save with **[Ctrl][s]**
10. Test that the code bank class and data files are correctly placed in Eclipse by running the program. Go to the class file “code.java” → **Right-mouse click**. → **Run as** → **Java Application**.
  - a. If correct, then you will see output in the Console window. Then log on as seen below.
  - b. If incorrect, then you will not see output in the Console window. Make correction(s) and redo.
11. Save a new class containing your template: **File** → **New** → **Class** → Name the class “**template**”.
12. Note (write) the path to the project folder for when you will upload the source code file \*.java

## *Logging On To The Specific Competition Page*

13. Navigate to the folder, "Friday Contests", then to the folder containing the day's current date.
14. Click on the dated shortcut to the contest. You should find yourself at USACO’s site.
15. Type your USACO UserName and Password. Choose the Name of Contest from the scroll down menu. Click the “Enter” button.

## ***Begin The Competition - Print & Copy Problems to Eclipse.***

16. Scroll the browser window to the **shortcut link** to the USACO Contest page containing the link to the questions to be solved. It will be the word “[English]” sitting to the immediate right of the name of the level of the competition that you will do. Example: “Novice division (Problems for first-year students) [English]” → The list of the problems will appear.
17. Print the web page containing the problems with **[Ctrl][P]** It is easier to read a problem on paper.
18. Highlight the USACO page containing the problems. → Highlight the problems and press **[Ctrl][c]**.
19. Go to Eclipse. **File** → **New** → Parent Folder: **21April2006** → File name: **Problems.txt**

## ***Set up the First Problems and It's Data File.***

20. Peruse the problems. Begin with the easiest problem.
21. Highlight name of the initial problem to be solved → Copy name with **[Ctrl][c]**.
22. Go to the Package Explorer. Highlight the name of the template file, “**template.java**”
23. Change all **strings** and **the related file name** from the template name to the problem name.

**Refactor** → Paste the name of the first problem to be solved. → New name: “**newProblem**” → Check **[✓]** Update references → Check **[✓]** Update textual matches in comments and strings (forces preview) → **Preview** → Left mouse click **Continue** → Left mouse click **OK**

24. Manually adjust the ID field of the Identification Header to your USACO handle

ID: **Your USACO handle** instead of “stud1”

25. Create the problem's input data file.

**File** → **New** → **File** → Select project folder as parent folder: **21April2006** → Use problem name + “in” for File name: **newProblem.in** → Finish → Go to **Problems.txt** → Locate **SAMPLE INPUT** → Highlight the sample data → copy with **[Ctrl][c]** → Return to the problem's input data file “**newProblem.in**”. → Paste data with **[Ctrl][v]**.

## ***Solve the Problem.***

26. Establish variables to read data from the input (\*.in) file. Know and use the Scanner class.

**Edit** → **Find/Replace** can be useful to replace “firstInt” with “n”, for example.

27. Delete template code that is immediately known to be irrelevant to solving the problem.
28. Copy and paste code from the code bank as applicable.
29. Test reading of the input data by mirroring it to the console with `System.out.println()` statement(s).
30. Remove mirroring `System.out.println()` statements after the reading of the input data is confirmed.

31. Replace all output to console with output to “newProblem.out”: **Edit** → **Find/Replace...** → Find: **System.out** → Replace With: **out** → **Replace all** → **Close**
32. Note: It is essential to conclude output with “out.close( );” to ensure the output buffer is emptied!
33. Open output file to verify output. → File → Open File... → Give path to project folder → Left mouse click name of output file “newProblem.out” → Open

### ***Submit Solution to USACO Robot.***

34. Navigate to the bottom of the USACO web page containing the problems.
35. Left mouse click the button labeled “**Browse**” → The “Choose File” dialog box appears. → In the window to the right of the “**Look in:**” label, navigate to the project folder → **File name:** → Left click the name of the output file “**newProgram.out**” → **Open** → Click the button labeled “**Submit**”
  - a. **If the solution is correct**, then the robot will return results indicating the compilation was “OK”, that all tests using different input data files were “OK”, and the amount of time the program took to execute with each test. See the following example.

```
PROB: prttri
LANG: JAVA

Compiling...
Compile: OK

Executing...
Sending test data
Test 1 OK [0.294 secs]
Test 2 OK [0.2886666666666667 secs]
Test 3 OK [0.2906666666666667 secs]
Test 4 OK [0.294 secs]
Test 5 OK [0.2933333333333333 secs]
Test 6 OK [0.2913333333333333 secs]
Test 7 OK [0.292 secs]
Test 8 OK [0.292 secs]
Test 9 OK [0.2946666666666667 secs]
```

- b. **If the solution is not correct**, then errors are noted as below. Repeat until successful.

```
Results for program 'prttri' (lang='JAVA'):

PROB: prttri
LANG: JAVA

Compiling...
Compile error: your program did not compile
correctly:
prttri.java:43: '.class' expected
    fo (int i=0; i<=n-2; i++)
        ^
```