
# Official Competition Manual

## March 2, 2005

Powered by [TopCoder]™

# Competition Instructions

## Logging In to the Competition Arena

Log in to the competition arena using the handle and password that you chose during the competition registration process.

Upon logging in, you will be placed into the general chat lobby. If you plan to compete in the competition round of the DoubleClick 2005 Coding Challenge, you will need to register for the event from within the competition arena during the 50-minute period prior to the start of the round. Ten minutes before the round starts, registration will close – be sure you do not miss registration. Register for the competition by doing the following from within the Competition Arena:

- Click on the "Active Contests" menu item (at the top, right)
- Select "DoubleClick Coding Challenge"
- Select "Register"

You can verify that you have registered correctly by checking the "Registrants" list in the "Active Contests" menu after you've registered.

Immediately following the close of registration the virtual "rooms" of competition will be created. When this process is complete, move to your assigned competition room as follows:

- Click on the "Active Contests" menu item (at the top, right)
- Select "DoubleClick Coding Challenge"
- Select "Enter"

You may use the above steps to get back to your competition room at any time during the competition.

## The Competition Room

From within the competition room, you will access the three problem statements that you will attempt to solve during the Coding Phase.

## Coding Phase

The Coding Phase is the period during which each contestant attempts to create solutions to three problem statements. The Coding Phase will last 75 minutes. When the competition begins, you may view the problem statements as follows:

- Click on the drop-down box labeled "Select one"
- Select the point value of the problem that you wish to solve

The higher the point value assigned, the more difficult the problem will be. You may open the problems in any order. You may have all problems open at the same time – but keep in mind that the timers for each problem will be counting down independent of other problems that may be open.

The Coding Phase will allow you to submit a problem more than once. If you've already submitted a problem and you choose to submit again, your score for that submission will be adjusted to account for the additional lapsed time, as well as an additional penalty of 10% of the maximum point value for the problem.

During the Coding Phase, the timer in the competition room window and the coding window will represent the amount of time left in the Coding Phase.

## The Coding Window

The coding window is the most important part of the Competition Arena. It is in the coding window that the problem statement is shown, and the solutions are coded, compiled, tested, and submitted.

### The Problem Statement

The top pane of the coding window contains the problem statement in its entirety. The problem statement explains exactly what it is that you are attempting to accomplish. It includes an explanation of the task, the class and method definitions that your solution must adhere to, all of the constraints for any input parameters, and examples to help clarify the statement.

### Choosing a Language

The DoubleClick Coding Challenge allows coders to utilize Java, C++, C#, or VB.NET as a competition programming language. You have the option to choose a programming language on a problem-by-problem basis. It is legal for you to use two different programming languages to solve different problems in a single competition. For this reason, you have the option of choosing the programming language from within the coding window. You can find the language selection area in the upper right area of the coding window – under "Choose your language". Remember to select the appropriate language before you attempt to compile your code. Strange compilation results can be a symptom of having the wrong language selected.

### Creating a Solution

The solution class you create and all of its members must be defined as public in order for the TopCoder servers to have access to your class members and invoke the appropriate methods. Below you will find code templates for Java, C++, C# and VB.NET:

#### Java Code Template

```
import <necessary classes>

public class <class-name> {
    public <return-type> <method-name>(<argument-type> arg1) {
        //your source here
        return <result>;
    }
}
```

## C++ Code Template

```
#include <necessary classes>

using namespace std; //Required for TopCoder gcc compiler

class <class-name>
{
    public:
    <return-type> <method-name> (<argument-type> arg1)
    {
        //your source here
        return <result>;
    };
};
```

## C# Code Template

```
using <necessary classes>;

public class <class-name> {
    public <return-type> <method-name>(<argument-type> arg1) {
        //your source here
        return <result>;
    }
}
```

## VB.NET Code Template

```
Imports <necessary-classes>

Public Class <class-name>

    Public Function <method-name>(arg1 as <argument-type>) As <return-type>
        --your source here
        Return <result>
    End Function

End Class
```

## Compiling

Code compilations from within the coding window will take place on TopCoder servers. The language that is chosen in the coding window at the time of the compilation will dictate to which language compiler the compilation request is sent. A compilation request will send the code that is currently in the coding window to the TopCoder compiler. If a compilation is successful, you will be notified of such. If the compilation is not successful, the errors that are returned by the compiler will be displayed to you in the compilation results window. Once a successful compilation is accomplished, the resultant compiled object will be stored on TopCoder's server. Any user test or code submission will always utilize the most recent successfully compiled code.

## User Testing

Once your code has been successfully compiled, you have the option of executing a user test against the compiled code. The user test will simply show you what your submission will produce for a given set of input parameters that you provide. A user test **will not** provide information as to whether or not your submission is producing the correct result. You will need to make your own determination as to the correctness of the return result.

Performing a user test based on the examples in the problem statement will provide a fairly good measure of the correctness of your solution since the examples will specify what the return value for a given set of input parameters should be. However, the examples do not represent an exhaustive test suite for a given problem – just because your code produces the proper results for all examples does not necessarily mean that it will produce the proper result for other input parameters. Think about potential border cases that may cause variations in the behavior of your solution and test these cases thoroughly.

If your compiled code executes successfully (without errors) for the input parameters that you've provided, you will be presented with the return value(s) that your code produced. In the event that an error occurs during the execution of your code, you will be presented with the error message(s) that the execution of your code produced.

***Important note: All code submissions must run in under eight (8) seconds for ALL test cases, otherwise the code submission will be marked as incorrect, and you will not receive any points***.

*NOTE:* any standard output (i.e., System.out.println) from the execution of your class/method will be returned to you along with the results of your test. This may be useful for debugging your code.

## The Problem Arguments Window

When you click on the "Test" button from within the coding window, the "Problem Arguments Window" will appear. The problem arguments window is where you will input the test case with which you wish to use to test your code. Each argument to the problem statement will be represented by either an input area (simple data type), or by a "Create" button if the argument is an array (complex data type).

## Simple Data Types

For all non-array data types - such as int, long, char and String - the test box input dialog will present you with a field to input the data. The data should be input without any quotes or other extra characters. So, if you want to test with the String "abc" you should just type abc into the input field, not "abc".

## Complex Data Types

When one of the inputs is an array such as int[] (vector<int> in C++), the test case dialog will present you with a button to create the array. Clicking this will bring up a new dialog box where you can enter the data elements. There are a couple of ways to enter data here. The most obvious is to enter the elements of the array one at a time and click the '+' button (or press enter) after each one. You can move the elements up or down using the '^' and 'v' buttons, respectively. You can also remove items either one at a time, with the '-' button, or all at once with the 'C' button. If you want to modify an element that is already entered, you can double click on the element in the panel above the entry field and modify it.

Adding elements one at a time can be slow, so there are also two buttons that allow batch adding of elements. The simpler of the two is the "++" button. To use this button, you

should enter all of the data elements as a comma delimited list, and then press the "++" button. For example, typing 1,3,6,4 and pressing "++" will create the array {1,3,6,4}. This button works exactly the same with all types of arrays. The other batch add button is the "{}" button. To use this, you should enter the data exactly as you would in your source code. So, to input {1,3,6,4} you would enter {1,3,6,4} and click "{}". While this may seem extraneous, given the "++" button, it has the advantage that it will parse String arrays that use double quotes, so that you can copy and paste the example test cases directly from the problem statement. For example, to enter the String[], {"a","b","c"}, you would simply have to enter {"a","b","c"} and press "{}". If you want to use the characters " or \ in a String[], you have to escape them with a \. Thus, \" represents a double quote, and \\ represents a single backslash. For example {"\"\\\""} represents a String[] with the single element: "\". The advantage to the "{}" button is that any array sample input can be copied directly from the problem statement, and created by pasting and clicking "{}". A couple of notes about the "{}" button are that the curly brackets at the front and end are optional, and the button will work without them. Also, whitespace that is not enclosed by double quotes is ignored. Furthermore, the dialog works exactly the same with all types of arrays. Thus, with the "{}" button {1,2,3} will create an int[],{1,2,3}, if the data type is an int[], and will create a String[], {"1","2","3"}, if the data type is a String[].

## Submitting

At any point after a successful compilation, you have the option to submit your code. It is important to realize that submission can be made regardless of whether or not your submission is correct. ***The faster you can submit a correct solution to the problem, the more points you will receive***. Points will be assigned regardless of whether the submission is correct or not. Any points for a submission that turns out to be wrong will be removed when the submission is subjected to the system tests. ***Submit will not necessarily use the code that appears in the coding window. You must compile first before you submit.***

You do have the option of submitting a problem multiple times. If you've already submitted a problem and you choose to submit again, your score for that submission will be adjusted to account for the additional lapsed time, as well as an additional penalty of 10% of the maximum point value for the problem.

## Saving

At any time while coding, you may save your code. Clicking the "Save" button will place the current version of your code on the TopCoder server. It is good to save every-so-often to ensure that a recent version of your code will be available in the event that something happens to your session (i.e., you lose your Internet connection). Compiling your code also has the effect of saving it.

## Clearing the Code

Using the "Clear" button, you can quickly clear out the coding window. Use this option only if you wish to start over on a particular problem. ***Warning: Once you clear your code, there is no way to retrieve it!***

### The Timer

The timer in the coding window will be always counting down to the end of the Coding Phase. Remember that there is variable latency between your computer and our servers. For that reason, it is ALWAYS best to perform any submissions with at least 30 seconds left on the timer.

## The Challenge Phase

The Challenge Phase will begin five minutes after the end of the Coding Phase (the period in between is an intermission), and will last for 15 minutes.

During the Challenge Phase, you have the opportunity to view the source code submissions of the other competitors in your room. If you believe that any of the submissions are flawed, you may challenge the submission with a specific test case that you feel will result in that submission returning the wrong result. If your challenge is successful, and the submission returns the wrong result, you will be awarded 50 points and the competitor will lose the points for the challenged submission. However, if your challenge is unsuccessful, and the submission returns the correct result for the test case, you will lose 25 points from your score.

To challenge a competitor's submission, open the summary window in the competition room by using the "Summary" button. Double click on any of the point values (shown in green) to view the source code for that submission. When viewing the source code, you will notice a "Challenge" button at the bottom of the window. The Challenge button will bring up the arguments window, allowing you to enter the test case with which to challenge the submission.

You may only submit a challenge if you have a positive (i.e., greater-than-zero) score at the time of the challenge. A given submission may only be successfully challenged once. If any competitor has already successfully challenged a submission, it may not be further challenged.

## System Testing Phase

After the Challenge Phase, all submissions will be run through a series of test cases. These test cases will compare an expected result against a received result to determine if the code submission works for each test case. ***Any submission that fails any test case will be deemed incorrect, and will result in a loss of all points for that submission. In addition, any submission that runs longer than eight (8) seconds for any test case will be deemed incorrect, and will result in a loss of all points.***

If your submission passes all of the test cases, you will keep the points that were assigned to you when you submitted the problem. Your final score for each round will cumulate the points remaining from the System Testing Phase and any points gained or lost during the Challenge Phase.

## If You Have Any Questions

If you have a question during the competition about the problem statement or about the functionality of the Arena, please utilize the chat interface in the Arena to present your question to the competition administrators. To do so, simply type the following in the white chat line at the bottom of the applet:

***admins: [YOUR QUESTION]***

An administrator will respond to your question in the chat area.