

In between retrieving and submitting problems, you can work on the local copy of your problem using whatever tools you wish. For example, in a C++ programming course, you may be able to use:

- a UNIX editor such as **nedit**, **emacs** or **pico**, a UNIX **g++** compiler, and the CourseMarker command-line (or GUI¹) client interface to retrieve the problems into directories in your UNIX account and later submit them. The installation in the school's UNIX system places the problems into directories in `~/CMhome/studentArea/yourlogin/`.
- the free **Dev-C++** development environment, which includes editor, compiler, debugger, and the CourseMarker GUI client interface to retrieve and submit the problems. The installation in the school's laboratories places the problems into directories in `H:\CMhome\studentArea\yourlogin`.
- your own development environment on your own PC, and the CourseMarker client interface to retrieve and submit problems. In this case, we provide the software but not much assistance. There is another document detailing the installation and configuration of CourseMarker client software.

In a Java programming course, you may be able to use:

- a UNIX editor such as **nedit**, **emacs** or **pico**, a UNIX **javac** compiler, and the CourseMarker command-line (or GUI) client interface to retrieve the problems into directories in your UNIX account and later submit them.
- the free **Netbeans** development environment, which includes editor, compiler, debugger, and the CourseMarker GUI client interface to retrieve and submit the problems.
- your own development environment on your own PC, and the CourseMarker client interface to retrieve and submit problems.

The flow sequence for using CourseMarker is something like that in Figure 2.

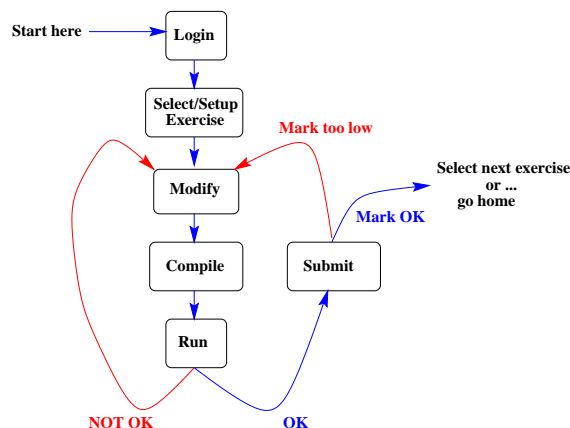


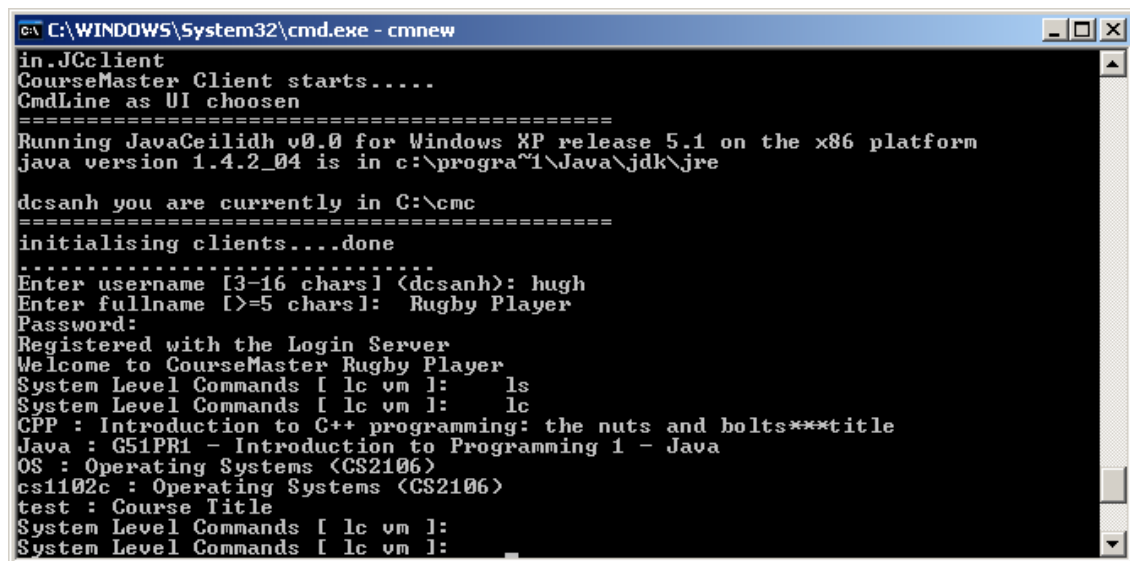
Figure 2: Flow sequence for CourseMarker use

¹Note that using the school's UNIX systems, and the GUI interface for CourseMarker with the X-window system works fine, but makes heavy use of the resources of the UNIX systems. It is much better to use the command-line interface if you are logged into the UNIX systems.

2 Using the CourseMarker command-line client interface

The CourseMarker command-line client interface is started by executing a command which is a batch file on Windows, or a script on UNIX. The name of the command is related to your course. For example if your course was CS1102, your command is `cm1102`. If your course was CS1101, your command is `cm1101`.²

When you first run the CourseMarker command-line client interface, it asks for your login, and a name before asking for your password. Later, the system remembers the name and some other configuration information.



```
C:\WINDOWS\System32\cmd.exe - cmnew
in.JCClient
CourseMaster Client starts....
CmdLine as UI choosen
=====
Running JavaCeilidh v0.0 for Windows XP release 5.1 on the x86 platform
java version 1.4.2_04 is in c:\progra~1\Java\jdk\jre

dcsanh you are currently in C:\cmc
=====
initialising clients...done
-----
Enter username [3-16 chars] <dcsanh>: hugh
Enter fullname [>=5 chars]: Rugby Player
Password:
Registered with the Login Server
Welcome to CourseMaster Rugby Player
System Level Commands [ lc vm l:      ls
System Level Commands [ lc vm l:      lc
CPP : Introduction to C++ programming: the nuts and bolts***title
Java : G51PR1 - Introduction to Programming 1 - Java
OS : Operating Systems (CS2106)
cs1102c : Operating Systems (CS2106)
test : Course Title
System Level Commands [ lc vm l:
System Level Commands [ lc vm l:
```

Figure 3: First login using command-line interface

In Figure 3, we see user `hugh` logging in, and supplying his full name³ “`Rugby Player`” and a password. The system will then display a selection of commands available at this level “[`lc vm`]” and await your command:

```
System Level Commands [ lc vm ]:
```

The `vm` (view message) command lists the message-of-the-day, and the `lc` (list courses) command lists the available courses, in this case:

```
CPP : Introduction to C++ Programming
Java : G51PR1 - Introduction to programming 1
OS : Operating Systems (CS2106)
cs1102c : Operating Systems (CS2106)
test : Course title
```

²On `sunfire` (UNIX) the scripts are found at `/home/course/cmaste/public`, so you would type `/home/course/cmaste/public/cm1102`. In the lab machines, the Windows batch files are found in `C:\cmc`, so you would type `C:\cmc\cm1102`

³The full name is only used so that CourseMarker can later say “`Welcome to CourseMarker Rugby Player`”. Silly.

You may then select one of the courses by typing its name (**CPP**, **Java**, **OS**, **cs1102c** or **test**), and a new selection of commands will be made available. The current set of allowable commands is always visible when CourseMarker is waiting for you to type in a command.

If you get this far, then that is great - go out and buy yourself some flowers. However, if things go wrong, have a look at the messages from the command⁴. It may be that

- a directory for the CourseMarker laboratory files cannot be found. Check for the **studentArea** directory.
- any connection to the CourseMarker server is failing. Check that you are not trying to use CourseMarker in a way expressly forbidden for your course. Some courses will restrict you to just use a single method of connecting. (i.e. you had been told that you *must* use the laboratories for CSXXXXX)
- the CourseMarker server is down. This is not likely, but you can ask.
- your installation is different in some way. If it is installed on your own machine, check that you have installed the software correctly, and that the paths in the **.bat** file and the configuration file are correct.
- ...your password is wrong...
- ...your usercode is wrong...
- ...the CourseMarker server does not like you...

2.1 Select exercise

```

C:\WINDOWS\System32\cmd.exe - cmnew
System Level Commands [ lc vm l]:  ls
System Level Commands [ lc vm l]:  lc
CPP : Introduction to C++ programming: the nuts and bolts***title
Java : CS1PR1 - Introduction to Programming 1 - Java
OS : Operating Systems (CS2106)
cs1102c : Operating Systems (CS2106)
test : Course Title
System Level Commands [ lc vm l]:  test
Course Level Commands [ vm csum lu up ]:  lu
ui : Unit Title
Course Level Commands [ vm csum lu up ]:  ui
Unit Level Commands [ lx vn usum up ]:  lx
ex1 : Single Celsius temperature to Fahrenheit
ex2 : Single Celsius temperature to Fahrenheit
ex3 : Hello World Testing
ex4 : Single Celsius temperature to Fahrenheit
Unit Level Commands [ lx vn usum up ]:  ex3
Exercise Level Commands [ set vq up ]:  set
Exercise Level Commands: [ set unset vq dev vtd vp up quit ]
Exercise Level Commands: [ set unset vq dev vtd vp up quit ]
Exercise Level Commands: [ set unset vq dev vtd vp up quit ]
Exercise Level Commands: [ set unset vq dev vtd vp up quit ]
Exercise Level Commands: [ set unset vq dev vtd vp up quit ]
Exercise Level Commands: [ set unset vq dev vtd vp up quit ]
Exercise Level Commands: [ set unset vq dev vtd vp up quit ]

```

Figure 4: Selecting exercise using command-line interface

You may then select an exercise to work on. In Figure 4, the **test** course is selected by typing its name, and we then get the “**Course Level Commands [vm csum lu up]:**” prompt. The **lu** (list units) command lists the available laboratory units (in this case just a single one, “**u1 : Unit Title**”).

We select this laboratory unit by typing its name “**u1**”, and we then get the “**Unit Level Commands [lx vn usum up]:**” prompt. The **lx** (list exercises) command lists the available exercises within the laboratory unit (in this case there are four exercises).

⁴In WinXX if you run a **cmd.exe**, and then run the script, you will see the messages in the window.

We select exercise 3 by typing its name, and get the “**Exercise Level Commands [set vq up]:**” prompt. The **set** (setup) command sets up this particular exercise, retrieving some files from the server and returning them to the client computer. At this stage we see that there are different commands available: “**Exercise Level Commands [set unset vq dev vtd vp up quit]:**”

If we now look around the disk on our local machine, we will see that various files have been placed in a special **studentArea** directory.

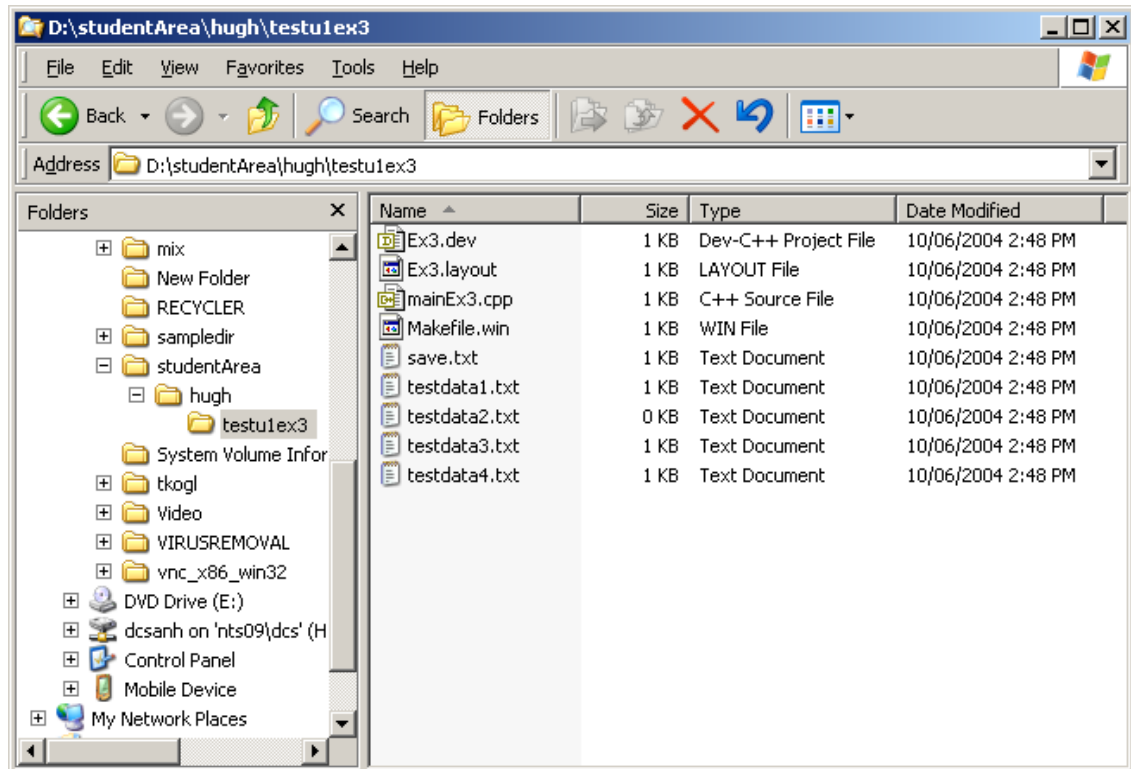


Figure 5: Files retrieved using CourseMarker

In Figure 5, we see that a new directory has been created, and it contains a series of files. In this case, the files are test data, special Dev-C++ project files and an initial **mainEx3.cpp** file.

There is also a **Makefile.win** file that is suitable for automating the compilation process under windows, but may need some modifications to work under UNIX. Typing the command “**make -f Makefile.win**” should do all the compiling and linking needed to build this program, leaving an executable file **Ex3.exe** in this directory. This executable is needed to signal to CourseMarker that it is possible to submit this exercise for assessment, even though the actual file submitted will be the source file **mainEx3.cpp**, rather than the executable file.

In UNIX, if you look in the directories **~/CMhome/studentArea/yourlogin/testu1ex3** you will find the files associated with the exercise. If you were using the WinXX environment in the laboratories, then if you look in the directories **H:\CMhome\studentArea\yourlogin\testu1ex3** you will find the files there. The directory name (**testu1ex3**) is constructed from the course (**test**), laboratory unit (**u1**) and exercise (**ex3**).

At this stage you should log out of the CourseMarker client (**quit**), and use your own tools to develop the laboratory. The next time you log in to CourseMarker, you will be in the same state that you just left, in the same exercise, in the same unit, in the same course.

2.2 Develop program

Let us suppose that we are using Dev-C++, and double-clicking on the project file for an exercise brings up the Dev-C++ development environment as seen in Figure 8.

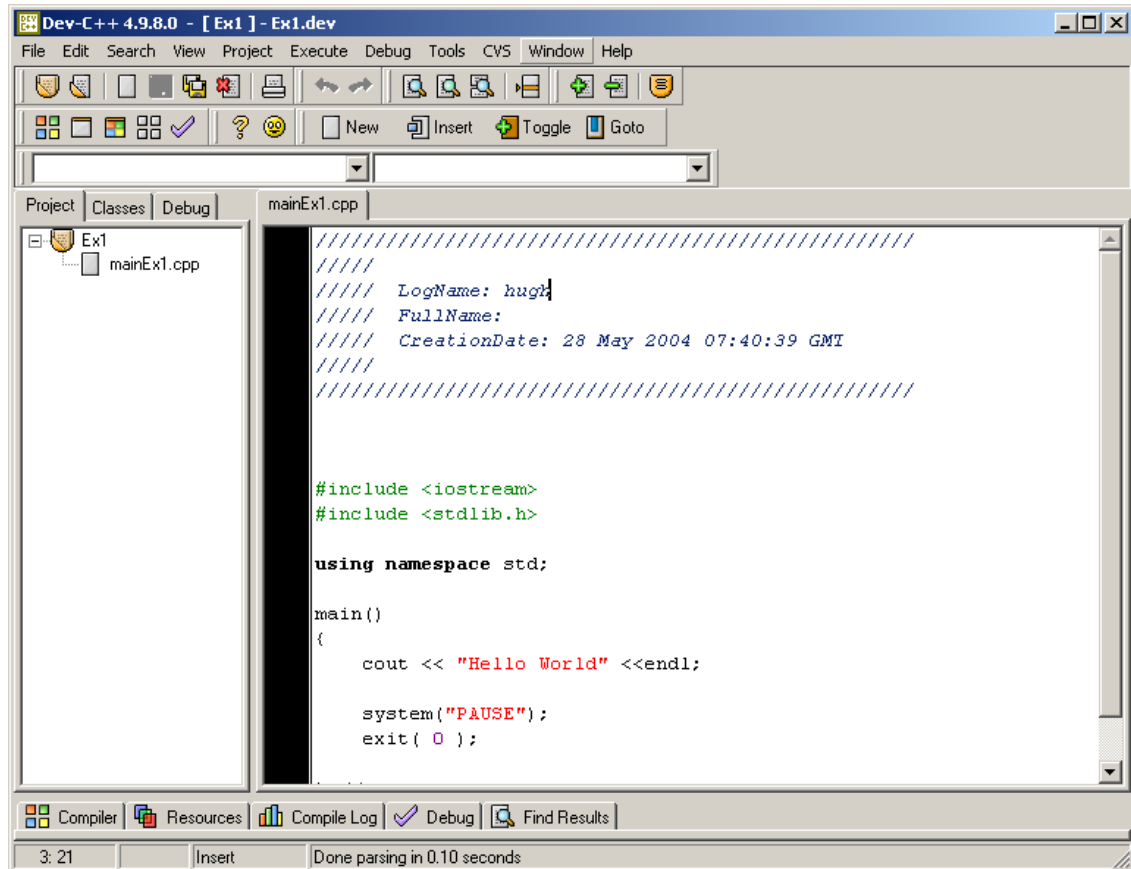


Figure 6: A development environment for C++

In this case, we can see an initial C++ program which has been downloaded `mainEx1.cpp`. Note that the initial program has been given a header, and some initial declarations.

You can modify the initial C++ source program `mainEx1.cpp` until you think it is correct, and then compile and run it using the menu items found under the *execute* menu. The `mainEx1.cpp` file is the one that will later be uploaded for assessment.

This particular development environment comes with lots of help available from the help menu.

2.3 Submit

Once the program appears correct, you may submit it to CourseMarker for evaluation by logging in again to the CourseMarker client software. When the prompt comes up, you will find that you now have new options: "**Exercise cmds [set unset vq dev vtd vp run rut submit vm up quit]:**".

If you cannot submit, then this means that there is not a compiled version of your program in the particular exercise directory. The CourseMarker client is pretty simple-minded. It looks for (say) `Ex1.exe`, and if it is there, it will enable submission.

Using the **submit** option sends the source of your program to the CourseMarker server. Depending on the exercise you may have anything from 1 to 10 submissions allowed, and in general the *best* submission will be the one used. When you **submit**, there may be a delay of from 1 to 30 seconds, depending on system load, and you will then get an assessment something like this:

```
(L1)=====
Grading: A
CWeight: 100
Description: Overall result
Feedback: Good result
(L2)=====
Grading: B
CWeight: 30
Description: Typography
Feedback: OK result
(L2)=====
Grading: D
CWeight: 20
Description: Dynamic tests
Feedback: Poor result
```

The result indicates that overall (at the outer level 1) an **A** was awarded, but the individual marks for the first two (level 2) components were **B**, and **D**. The figures next to the grades (**30**) (**20**) indicate the relative weighting of the assessments, in this case indicating that 30% of the assessment was for typography (the look and style of the source code) and 20% of the assessment was for a series of dynamic tests.

The process does not end there. Automated assessment is useful, but we do not rely on it. Your best submission will be examined by a tutor, who will make comments on your code, and provide further marks. Your final assessment will be mailed to you after the laboratory has closed, and all the submissions have been individually assessed.

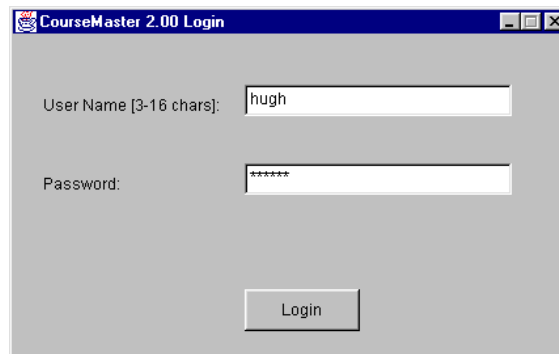
2.4 Commands for command-line interface

Level	Command	Comment
System	lc	List courses on this server
	vm	View the message of the day
Course	vm	View a short message related to this course
	csum	View a summary of information related to this course
	lu	List units
	up	Go up to system level
Unit	lx	List exercises
	vn	View notes related to this unit
	usum	View a summary of information related to this unit
	up	Go up to course level
Exercise (initially)	set	Setup this question, creating the needed directory
	vq	View the question
	up	Go up to unit level
Exercise (after setup)	unset	Removes this (previously setup) question. Dangerous!
	dev	Run the development environment
	vtd	View the test data
	vp	View the program name
	quit	Quit program
Exercise (after compile)	run	Run the program
	rut	Run the program against some set of testdata
	submit	Submit your answer
	vm	View the previous submission's marks

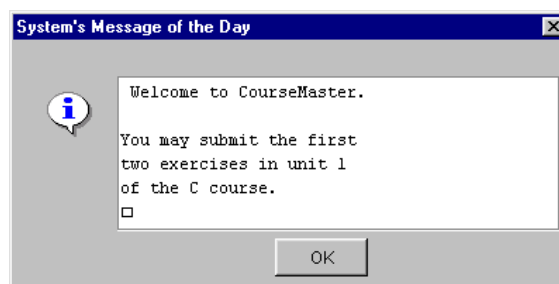
3 Using the CourseMarker GUI client interface

The CourseMarker GUI client interface is started by executing a command which is a batch file on Windows, or a script on UNIX. The name of the command is related to your course. For example if your course was CS1102, your command is **cm1102gui**. If your course was CS1101, your command is **cm1101gui**⁵.

When you first run the CourseMarker GUI client interface, you are presented with a login screen, in which you have to enter your CourseMarker account name and password:



The system will then display a Message-of-the-day in which we will put any important news (so you should read it)



If you get this far, then that is great - go out and buy yourself some flowers. However, if things go wrong, have a look at the messages from the command⁶. It may be that

- a directory for the CourseMarker laboratory files cannot be found. Check the **studentArea** directory.
- any connection to the CourseMarker server is failing. Check that you are not trying to use CourseMarker in a way expressly forbidden for your course. Some courses will restrict you to just use a single method of connecting. (i.e. you had been told that you *must* use the laboratories for CSXXXXX)
- the CourseMarker server is down. This is not likely, but you can ask.
- ...your password is wrong...your usercode is wrong...the CourseMarker server does not like you...

⁵On **sunfire** (UNIX) the scripts are found at **/home/course/cmaste/public**, so you would type **/home/course/cmaste/public/cm1102gui**. In the lab machines, the Windows batch files are found in **C:\cmc**, so you would type **C:\cmc\cm1102gui**

⁶In WinXX if you run a **cmd.exe**, and then run the script, you will see the messages in the window.

3.1 Select exercise

You may then select an exercise to work on. In Figure 7, the left window allows you to select a particular exercise for submission. The right window contains information about the course (if you have a course selected), the laboratory unit (if you have a unit selected), and if you have an exercise selected, it contains a description of the programming exercise that you have to do. Read all of these carefully.

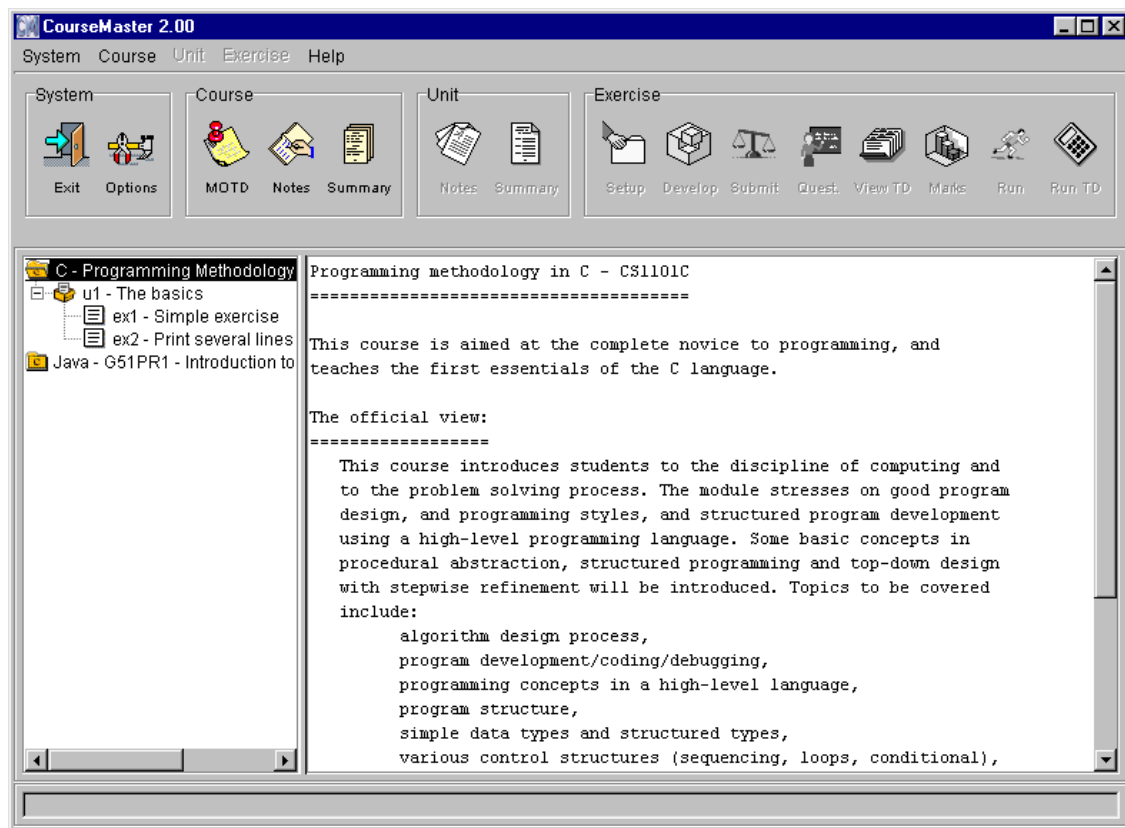


Figure 7: Selecting CourseMarker exercises

At times you may see other courses in the windows. This may be the administrators testing things, or there may be another similar course running on your server.

In Figure 7, you should be able to see a C programming course, and a Java programming course in the left window. If you look more closely, you will see that the C programming course has a single laboratory available, labelled “u1 - The basics” (for unit 1). Within that laboratory, there are two laboratory exercises: “ex1 - Simple exercise”, and “ex2 - Print several lines”.

The first time you select an exercise, click the *setup* button and an initial set of files will be downloaded to your machine. If it downloads correctly, you will get notification. The programming exercise you are to work on is now stored on the machine you are working from. If you were using the UNIX environment, then if you look in the directories `~/CMhome/studentArea/yourlogin/Culex1` and `~/CMhome/studentArea/yourlogin/Culex2` you will find the files associated with each exercise. If you were using Windows, then look in the directories `H:\CMhome\studentArea\yourlogin\Culex1` and `H:\CMhome\studentArea\yourlogin\Culex2`, and you will find the files there.



3.2 Develop program

At this stage you have a choice. You can

- log out of the CourseMarker client, and use your own tools to develop your laboratory, or
- keep the CourseMarker client running, and use the tools from its menu⁷.

Let us assume that you are running from Windows, so you leave the CourseMarker client running, and then click the *develop* button. At this point a development environment of some sort runs. In the case of a C++ course, Dev-C++ will run as seen in Figure 8.

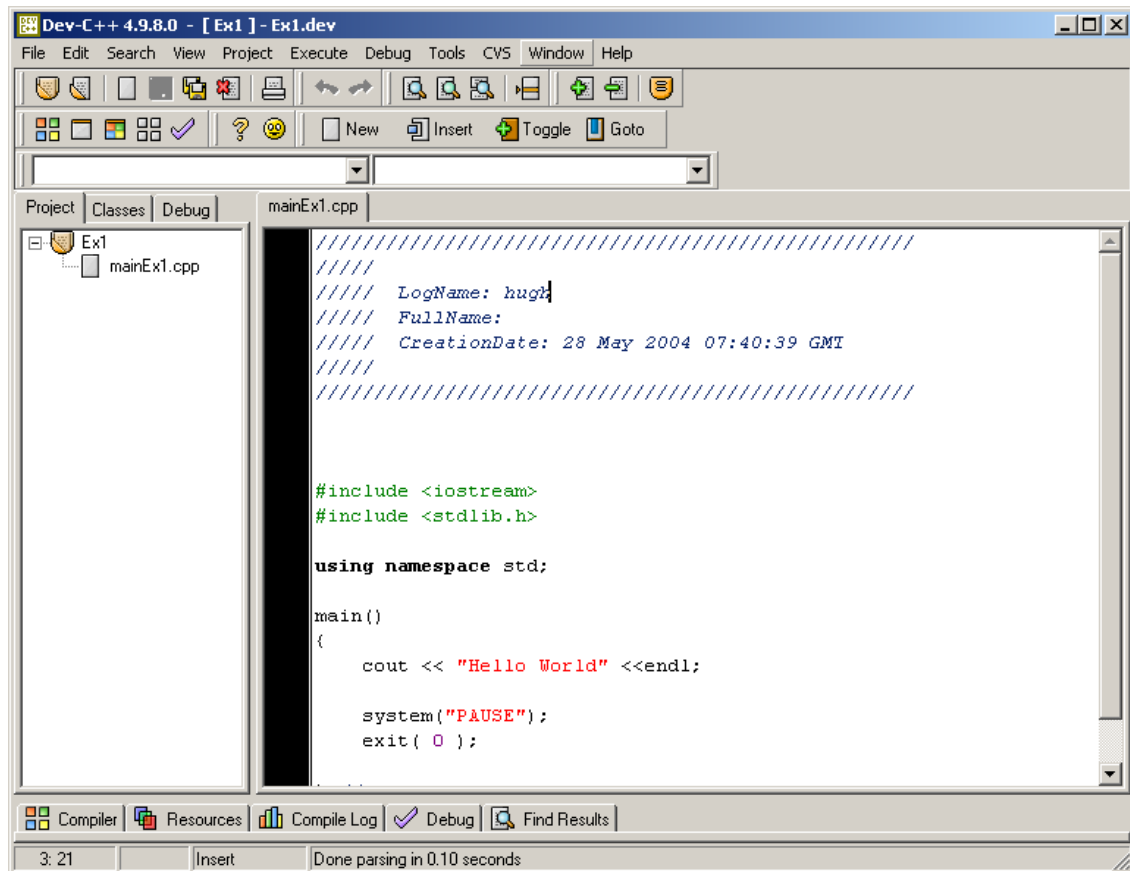


Figure 8: A development environment for C++

In this case, we can see an initial C++ program which has been downloaded `mainEx1.cpp`. Note that the initial program has been given a header, and some initial declarations.

You can modify the initial C++ source program `mainEx1.cpp` until you think it is correct, and then compile and run it using the menu items found under the *execute* menu. The `mainEx1.cpp` file is the one that will later be uploaded for assessment.

This particular development environment comes with lots of help available from the help menu.

⁷On WinXX, we encourage you to leave it running, and use the tools provided in the laboratories, and accessible from the menu. On UNIX we encourage you to exit the GUI as quickly as possible.

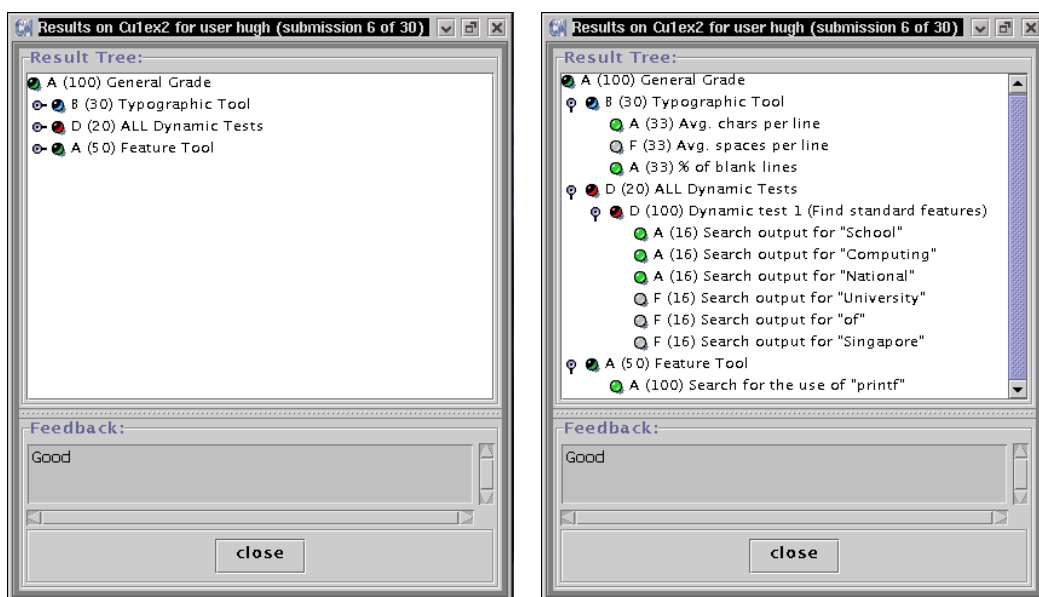
3.3 Submit



Once the program appears correct, you may submit it to CourseMarker for evaluation using the *submit* button which sends your program to the CourseMarker server. Depending on the exercise you may have anything from 1 to 10 submissions allowed, and in general the *best* submission will be the one used.

If the submit button is not highlighted, deselect the exercise and then reselect it. If it is still not highlighted, then this means that there is not a compiled version of your program in the particular exercise directory. The CourseMarker client is pretty simple-minded. It looks for (say) `Ex1.exe`, and if it is there, it will highlight the button.

Once you can submit, your source code will be copied from your directory, and copied to the server, where it will be assessed online. Within a short time you should get the result of your submission.



(a) Overall view

(b) Detailed view

Figure 9: Results of submission

The result in Figure 9(a) indicates that overall an **A** was awarded, but the individual marks for the components were **B**, **D** and **A**. The figures next to the grades (30) (20) (50) indicate the relative weighting of the assessments, in this case indicating that 30% of the assessment was for typography (the look and style of the source code), 20% of the assessment was for a series of dynamic tests, and 50% of the assessment was for including some requested features in your program.

You can examine the results in more detail, by clicking on the little controls next to the grades. Figure 9(b) indicates more detail, and we can see the results of six dynamic tests. If you click on each assessment, you may get further comments.

The process does not end there. Automated assessment is useful, but we do not rely on it. Your best submission will be examined by a tutor, who will make comments on your code, and provide further marks. Your final assessment will be mailed to you after the laboratory has closed, and all the submissions have been individually assessed.

4 Installation of CourseMarker

The installation of GUI on your own PC is a little complicated but is a convenient way for you to access the lab sessions from home. The brief lecture on CourseMarker is not really enough to get the job done, so the following explanation has been developed by engineering student Qin Tian:

1. **Download CM:** Download the CM software from the SoC course web site. Downloaded file is a zip file. Unzip to `D:\cmc`. (Please do unzip it to `D:\cmc`, I can't guarantee other directories will work). Double check when you finish unzipping, because some unzipping software automatically establishes a new directory under the one you specify, which may lead to an unsuccessful installation. So make sure you have all your subdirectories and `.bat` files under `D:\cmc`, and not `D:\cmc\cmc`, otherwise, cut them and paste them into the upper level.
2. **Decision about Java:** You need (at least) the JRE (Java runtime environment) to run CourseMarker client software. We assume that you are using JDK (Java developer's kit) version 5.0, revision 1⁸, and there is a copy available at the SoC course web site. If your course is a Java programming one, then you need the JDK, not just the JRE.
You might have it already, or you might have an older version, and a problem might be that your JRE/JDK may be installed under a differently named directory. It may be possible to use your existing JRE/JDK, when you edit your configuration file (something like `cm1102c.bat`) in the following step. Otherwise, download and install the latest copy of JRE or JDK.
3. **Adjust configuration:** At NUS, the `H:` drive is used for your home NUSNET directory, and the CM software in the laboratories (by default) stores your files there. The CM software that you have just downloaded assumes that you want to store your files on the `D:` drive of your PC. If you already have a `D:` drive on your PC you can skip step (b) below:
 - (a) Enter into `D:\cmc` where you unzipped your software. Right mouse click on the batch file for your course (say `cm1102gui.bat`), and click "edit". You will see a text file like document with lots of information in it. (The text begins with `ECHO ON...`).
 - (b) OK, now change the "`set CMHOMEDIR=D:`" you see to the drive name you want to use. For example such as "`set CMHOMEDIR=C:`". (If you have `E` disk, then `E` will do as well).
 - (c) You see the third line "`set JAVA_HOME=...`"? Edit the right hand side of the equation, change it to the directory into which you just installed your JRE/JDK. WARNING! You *must* use short names for the directory. You can see the short names by using the command "`dir /x`" in a CMD window. The default directory may be `C:\Program Files\Java\jre1.5.0_01`, but you must write `C:\Progra~1\Java\jre15~1.0_0`. Save your changes and close the file.
4. **Test:** Run the batch file (`cm1102gui.bat`) to enter the environment. I won't explain how to use the software. It's easy. After setting up the exercises, if everything goes well, you will find your exercises stored in the directory you specified previously, either `C:\CMhome`, `D:\CMhome` OR `E:\CMhome`.

Note: If you are using Dev-cpp for a C++ course, and the "Develop" button doesn't work for you, then your Dev-cpp may have been installed into other directories by you. Edit the file `rundev.bat`.

Note: If you are using Java, then by default the file `rundev.bat` uses a simple editor called PFE, which is supplied along with CM. If you are using this, then double-click on the file `pfemenu.reg` found in the directory `D:\cmc\pfe`. This adds the java compiler and other items to the menu.

OK, I guess that will be what you need to do. There might be some steps you don't need to do, and some are not crucial as there will be other alternatives, but follow the steps, and you will at least get your CM to run.

⁸But older copies may work as well...