CMSC 330 Project submission and grading policies Fall 2012

1 Grace account setup

Programming will be done on the OIT Grace Cluster, which may be reached at grace.umd.edu. You will use your own TerpConnect account to access the Grace cluster; your login ID and password are just your University directory ID and password. Below are steps to set up your account for this course, including procedures to check that things were done correctly.

Every student gets extra disk space during each semester that they are taking class using the Grace systems. It's **extremely important** to do all your programming work for this course in this extra disk space, rather than your home directory, for two reasons:

- You won't need to worry about running out of disk space in your home directory, even if you accidentally create some very large files, and
- The extra disk space is automatically backed up every night, so if you were to accidentally erase an important file, the previous day's version can be recovered.

The extra disk space is located elsewhere on the system (not in your home directory), but it's easy to create a symbolic link pointing to it, as explained below. (Note that the course extra disk space from CMSC 216 should **not** be used, because even if it is currently accessible it will become inaccessible during the semester, and you will not be able to use any files there after that time.)

As mentioned in the syllabus, although programming may be done on other systems that you have access to, it is recommended that you do all your coursework on the Grace cluster. Note that the Eclipse IDE can also be used to do programming, but you are encouraged to use the Grace systems (or at least some other UNIX/Linux system). Experience with more than one operating system, not just an IDE, is invaluable for any computer scientist or engineer. Some practice in the UNIX development environment can greatly increase your productivity in it. If you do use any other system for programming, keep in mind that projects must work correctly on the Grace machines (and the CMSC project submission server), and no consideration in grading can be made for errors made in transferring files, or submitting the wrong version of a project.

In a couple of steps below you have to add a line to one of your account dotfiles (or account control files); if you add a line at the bottom, be sure it ends with a newline, or it won't be recognized by the shell, so just press enter at the end of typing the last line of any file.

1. Log into grace.umd.edu, using your UMCP directory ID and password, and use a text editor to edit the file `.path` located in your home directory. (Emacs was covered in CMSC 216; other text editors may be used, but `pico` is not recommended as it is too simple to be used for programming.) Add the following line to the bottom:

   ```
   setenv PATH "/afs/glue/class/fall2012/cmsc/330/0101/public/bin:${PATH}"
   ```

   Be sure that `${PATH}` appears after the other, long directory name as shown, and that the curly braces are present. Check this using the following commands:

   ```
   source ~/.path
   echo $PATH
   ```

   You should see `/afs/glue/class/fall2012/cmsc/330/0101/public/bin` as the first component of your path, which is where the project submission program is, and other programs as needed will be placed.

   Note: if you already have a line like the above at the end of your `.path` file from CMSC 216 just remove it, as that directory is no longer accessible, or will become inaccessible during the semester.

2. Project test inputs and outputs, discussion section example code, and other files, will be provided in the public class directory, which is:

   ```
   /afs/glue/class/fall2012/cmsc/330/0101/public
   ```

   However, since using this path is a lot of typing, just create a symbolic link to it from your home directory instead. To do so, cd to your home directory, if you're not there already, and use the command:

   ```
   ln -s /afs/glue/class/fall2012/cmsc/330/0101/public 330public
   ```

   This will create a symbolic link named "`330public`" to the public class directory. Check this using the commands:

```
cd 330public/..
pwd
```

This should print the path `/afs/glue/class/fall2012/cmsc/330/0101`.

Note: if you already have a symlink `216public` from CMSC 216 in your home directory just remove it, as that directory is either no longer accessible, or will become inaccessible during the semester.

3. Similarly, to create a symbolic link to your extra course disk space, cd to your home directory if you're not already there, and give the command:

   `ln -s /afs/glue/class/fall2012/cmsc/330/`*section*`/student/`*loginID* `330`

   where *section* and *loginID* are to be replaced by your own discussion section and TerpConnect login ID. Check this using the commands:

   ```
   cd 330/..
   pwd
   ```

   This should print the path `/afs/glue/class/fall2012/cmsc/330/0101/student-cmsc330-`*section*, where your own section number will appear in place of *section*.

   From now on, anytime you log in to do any type of coursework for this class, be sure to first just "`cd 330`", to change to your extra course disk space, so that all your programming is done **in this extra disk space**.

   Note: if you already have a symlink `216` from CMSC 216 in your home directory just remove it, as the extra course disk space from prior semesters is either no longer accessible, or will become inaccessible during the semester.

4. It's necessary to be able to use Oracle Java, which is not the default version on the Grace systems, but this can easily be changed. Add a line that just reads "`tap -q java`" (without the quotes) to your file `.cshrc.mine`: Check this using the commands:

   ```
   source ~/.cshrc.mine
   rehash
   which java
   ```

   This should print `/afs/glue.umd.edu/software/java/current/sys/bin/java`, not `/usr/bin/java`.

2 Project submission, deadlines, and grading policies

Projects will be submitted electronically to the CMSC project submission server, and directions will be provided with project assignments. Projects submitted via any other means (such as an emailed project) cannot be considered. **Do not** submit projects using the submit server's mechanism for uploading a jarfile or zipfile. Use the procedures to be described with assignments. Projects may fail on the submit server if a zipfile or jarfile is submitted, and allowances cannot be made in grading for problems with submissions made in this manner. Only the projects electronically submitted according to the procedures provided can be graded; it is each student's responsibility to test their program and **verify that it works properly** before submitting, as well as to **log into the submit server** and verify that their submission worked correctly there.

All projects will be due at 10:00:00 p.m. on the day indicated on the project assignment. Projects may be submitted up to two days late, with a 15–point late penalty for each late day. If you submit more than once, you will receive the highest score of your submissions, meaning the highest score after the late penalty is taken into account for any late submissions. Submission deadlines are **firm** and exceptions cannot be made. Note there is **no grace period** for project submissions– deadlines are enforced by the submit server at exactly 10:00:00 p.m., to the second, the day a project is due, and every 24 hours later for one-day–late and two–day–late submissions. Note that project scores as they appear on the project submission server will not correctly include late penalties in all cases, however we will incorporate these into project grades when they are entered electronically.

Project extensions cannot be given to individual students as a result of system problems, network problems, power outages, etc., so do not leave finishing and submitting a project until the last minute. It is strongly suggested you finish and submit your program at least one day early, to allow time to reread the project assignment and all relevant articles in the course Piazza space (see below), to insure you have not missed anything which could cause you to lose credit on the project.

Projects will not be graded on style or documentation, although a certain design or structure, or certain language features, may be required for some projects.

Some public test inputs and outputs will be provided for projects, but note that unlike lower–level programming classes, we will not be providing extensive automatic testing before projects are due, and projects will be graded largely based on test cases **not provided in advance**. Instead you will be responsible for developing your own techniques for testing your own projects, and for checking the correctness of your output yourself. Similarly, projects will not have any release tests.

Note that any hardcoding in a project assignment will result in a score of zero for that project. Hardcoding refers to attempting to make a program appear as if it works correctly and actually calculates and computes correct results, when for some reason it actually does not do so. A few examples would include a program which prints the desired output instead of computing it, or a program that works only because it takes advantage of properties that the provided test inputs happen to have, etc. Hardcoding may be considered to be a violation of the University's academic integrity policies, as it may be construed to be fabrication or falsification of information, so if you have any question about whether a particular situation would constitute hardcoding be sure to ask ahead of time.

3 General project advice

- Be sure to carefully read Section 7 of the course syllabus, regarding academic integrity.
- Be sure to save your work in the text editor often, regardless of what computing platform you are using. A power failure, network failure, or dropped connection could otherwise cost many hours of lost work.
- Be sure to keep backup copies of your project and any other important files (like your own test data) in a different directory or a subdirectory of the directory where your project is located. It's a good idea to save backup copies every time you log in or log out, for example.

  Don't use the submit server to keep backup copies of your program, as having a large number of unnecessary submissions just slows things down on the submit server for everyone.
- Don't just rely on the submit server to check whether your program works on any public test inputs, meaning before submitting you should run it yourself on any public test inputs, and compare its results yourself.

4 Piazza

Everyone registered should have received information via email by now about joining the course Piazza space. This may be very useful for asking and answering questions about projects and other coursework. You should check it often, once programming assignments have been made, to see if any messages there are relevant to the current project. Please read and keep the following in mind these rules regarding Piazza:

- You may post or reply to questions about project requirements, language features, general syntax errors, or course material via Piazza. However, you **may not** ask or explain how to implement any part of a project in a public venue (or privately, for that matter). Such questions can be asked in person during office hours. Asking or explaining how to implement any part of a project via Piazza would be a violation of this course's academic integrity requirements.
- Anonymous messages in Piazza may be deleted. Let your fellow students know who is asking and answering questions.
- Piazza should be used for communication regarding class material. As it will be read by all the students in the course (in addition to the instructional staff) please do not make everyone read through unnecessary messages to find information they need for their coursework. In particular, don't post any messages which have no relevance to the course material. There are many other venues in which matters unrelated to the course (or not of interest to students in the course) may be discussed.
- It is not uncommon that questions are asked in a venue like Piazza that are actually specifically addressed in a project assignment; although we don't want to discourage questions, to reduce the number of messages that everyone must read we ask that you first check a project assignment before posting a question about it Similarly, questions may be asked that have already been answered, so first read any pending unread messages prior to posting questions in Piazza.
- Piazza cannot be used to ask questions about anything particular to one student. If you have any questions or concerns or concerns about the course that are not related to the content of the course material you are encouraged to discuss them with your instructor in person, during office hours, or before or after class.