Computing & Software Systems 332: Programming Issues With Object-Oriented Languages Spring 2014

Basic Course Information

Covers language and development/execution environment differences, including data types, control structures, arrays, and I/O; addressing and memory management issues including pointers, references, functions, and their passing conventions; object-oriented design specifics related to structured data and classes.

Instructor. Michael Stiber stiber@u.washington.edu, room UW1-360D, phone (425) 352-5280, office hours Mondays 4–5:30PM or by appointment.

Course Web. `http://courses.washington.edu/css332/stiber/`

Class Meetings. Fridays, 1:15–5:15PM, UW1-051.

Course Objectives. The overall goal of CSS 332 is to transition students who are competent developing small programs in one object-oriented programming language to developing larger programs another object-oriented language. Currently, it is assumed that students are familiar with Java and need to become familiar with C++, the language used in CSS 342/343. While the underlying concepts are similar, there are some differences in syntax and many differences in usage, particularly relating to memory management and the software development toolchain. This course is intended as an aid to your survival in CSS 342. The intention is to bring Java students up to speed in C++ as quickly and painlessly as possible. Consequently, this course is dense, taught concurrently with the first half of CSS 342. The intention is that the first programming assignment given in CSS 342 (review) is in conjunction with work in CSS 332. There may be additional in-class work that focus on specific aspects of C++. This course will be run as part lecture, part hands on lab.

Suggested Reading and References.

- Bruce Eckel, *Thinking in C++*, Second Edition, vols. 1 & 2, Prentice Hall, 2000. A good "from scratch" introduction to the language. Available for free in electronic form on-line at `http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html`.
- Bjarne Stroustrup, *The C++ Programming Language*, Third Edition, Addison-Wesley, 2000. The canonical reference. Make sure you get the third edition.
- Harley Hahn, *Harley Hahn's Student Guide to UNIX*, 2nd edition, McGraw-Hill, 1996. If you're unfamiliar with Unix or Linux and want to learn more, get a book like this.
- Herbert Schildt, *STL Programming from the Ground Up*, Osborne/McGraw-Hill, 1999. A very good introduction to the Standard Template Library, with lots of examples. However, it is not a reference; for example, it doesn't provide complete lists of methods for each class.
- Nicolai M. Josuttis, *The C++ Standard Library*, Addison-Wesley, 1999. Much more of a complete reference than the Schildt book. Includes some examples, but is not intended as a tutorial.

Grading. This class is graded as C/NC. There are no exams, but programs will be written and reviewed during class. You must attend and participate seriously in lab activities to receive credit.

Special needs

If you believe that you have a disability and would like academic accommodations, please contact Disability Support Services at (425) 352-5307, TDD (425) 352-5303, FAX (425) 352-5455 or at dss@uwb.edu. In most cases, you will need to provide documentation of your disability as part of the review process. More information is available at the DRS web site.

Student Conduct; Plagiarism and Cheating

You are expected to provide original work based on your own effort for this course. You will receive a zero for any coursework for which you are discovered cheating or plagiarizing. You will be referred to the University for further action. It is your responsibility to know and uphold the Student Conduct Code for the University of Washington, available at `http://www.uwb.edu/students/policies/`

Problems

If you have problems with anything in the course, please come and see me during office hours, make an appointment to see me at some other time, or send email. I want to make you a success in this course and, by extension, CSS 342. If you have trouble with the assignments, see me before they are due. If you fall behind, it will be difficult to catch up.

Course Schedule

Generally speaking, class activities will be coordinated with topics and assignments being covered in CSS 342. This may not mean *exact* coordination, as we will be covering some practicalities that CSS 342 doesn't cover and because this course only lasts half of the quarter. Topics (and their rough estimated dates) to be covered will be:

April 4 The C++ toolchain, including preprocessor, compilation, linking, and debugging in Windows, Mac OS, and Linux. Debugging practicalities, such as `cerr`, conditional compilation and using a `DEBUG` definition. Primitive type conversion (implicit and explicit). Pass by value and pass by reference. `struct`.

April 18 Pointers, references, address-of operator, memory allocation and deallocation. Difference between `null` and zero. Relationship between arrays and pointers. Copy constructors. Deep vs. shallow copy. `valgrind`.

April 25 Heap and stack. Identifier scope; namespaces. Object lifespan and `static`. Lvalues and `const`. Overloading built-in operators, both as methods and free functions. Stream I/O, `cstdio`, and differences in text I/O between different operating systems.

May 2 Structures and C++ classes, header files, exceptions, memory management in constructors and destructors. The command line.

May 16 Pointer arithmetic. Pointers to functions. Memory management revisited. Operator precedence and derived types. `void`. Clearing memory upon allocation and before deallocation. `cmath`.

Additional Topics, time permitting: C++ inheritance implementation. Implicit and explicit type conversion for classes. Enumerations. Bit fields. Unions. Iterators and templates.