MET CS665 Design Patterns and Component Software

Boston University Metropolitan College

Fall 2014
PSY B41
Monday 6:00pm – 9:00pm

Course Description

This course is an introduction to software design patterns. A design pattern is a general reusable solution to a commonly occurring problem within a given context in software design. Patterns are formalized best practices that the programmers should implement in the applications. The course covers the most common object-oriented software design patterns. For each pattern, it will discuss the underlying object-oriented design principle it is based on; the specific object-oriented design problem it solves; the context it can be applied and the consequences of applying it; how the different components of the pattern collaborate with each other; and etc.

Course Prerequisites

Student should have solid background on object-oriented programming. MET CS 341 or 342 (Data Structures with C++ or Java) is required for understanding the contents of this course. MET CS 565 (Advanced Java Programming) is highly recommended, but it not required here. The course is emphasizing the object-oriented design patterns concepts independent of programming language. All examples in class are using Java programs. But students have the choice of using either Java or C++ to implement the homework or project assignments.

Textbooks

Required

Head First Design Patterns
By Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra
O'Reilly, 2004

Recommended

Design Patterns: Elements of Reusable Object-Oriented Software
By Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides
Addison-Wesley, 1994

Course Web site

The course Web page is hosted on Blackboard Learn (https://learn.bu.edu). The class notes, homework assignments, and additional announcements will be posted here.

Instructor

Jie Lu
jielu@bu.edu
However, I prefer using the Mail tool in Blackboard for communication regarding this course.

Attendance Policy

Class attendance is important. Individual student is responsible for all scheduling and other announcements made in class. The likelihood of failing the course is subsequently increased if one fails to attend class regularly.

The format of this course will be lectures, which will include working through sample problems. Certain course material will only appear during lectures, and most announcements will only be made in class.

Grading Criteria

Grades will be based on your participation in class discussions (10%), your grades for homework exercises (50%), and the final examination (40%).
Grading criteria for homework include: (a) correct implementation of the specified design pattern, (b) complete and non-redundant testing, (c) clearly and correctly written description of the design and testing and (d) correct use of UML or other notation.

Academic Honesty

The course is governed by the Boston University Academic Conduct Committee policies regarding plagiarism (any attempt to represent the work of another person as one's own). This includes copying (even with modifications) of a program or a segment of code or documentation or letting your work to be copied. Your participation in interactions with the instructor and your classmates is encouraged, but the work you submit must be your own. Collaboration is not permitted ((for

more details, see http://www.bu.edu/met/for-students/met-policies-procedures-resources/academic- conduct-code/).

Computing Facilities

As a Boston University student, you have the right to use the bookstore, libraries, sport facilities, and parking. The same is true of Boston University computing facilities. If you wish to use a Boston University Academic Computer Center account, one will be created for you, and you will be able to use its Unix operating system and its Java compiler. The Center office and the terminal room are located at 111 Cummington Street and in several other locations (to open an account, you have to show a valid BU ID card and choose your user name and a Kerberos password). System help and information is available at 617-353-2784.

You can also use the MET College computing labs that include PCs running Windows, Linux, or Unix (808 Commonwealth Avenue, 2nd floor). To enter the lab, you should register your BU ID card with an operator. To use a PC, you log in using your BU user name and your Kerberos password.

This course, however, is not environment-specific. Those students who have access to other computer systems that support Java should feel free to use them. Make sure, however, that these systems run the Java (tm) 2 Platform Standard Edition (any version).

If you want to download the Java Development Kit to your personal computer, go to the Sun Microsystems web site. There is no need to go after the Enterprise Edition (J2EE), the Standard Edition (Java SE) is sufficient. Download the Java SE Development Kit (any version) from http://www.oracle.com/technetwork/java/javase/downloads/index.html. To download API documentation, click the Documentation tab on that page.

Notice that you do not need to learn how to use a debugger or an IDE (Integrated Development Environment) for this course. Learning these tools will significantly increase the number of problems you will have to deal with in this course. Meanwhile, the programming problems in this course are simple enough and could be written using a simple text editor. They should be debugged and tested using your intelligence, not sophisticated and complex software tools.

The recommended development environment is TextPad from http://www.textpad.com. Its trial version is free, its user interface is very simple and intuitive, and it is sufficient for the course work. Make sure that you install Java JDK first and Textpad next to enable Textpad to find your version of Java.

Course Calendar

Week	Date	Topic	Reading
1	09/08	Course introduction; Object-oriented design basics and principles	Notes
2	09/15	Introduction to design patterns; The Strategy Pattern	1
3	09/22	The Observer Pattern	2
4	09/29	The Decorator Pattern	3
5	10/06	The Factory Pattern	4
6	10/14	The Singleton Pattern (Tuesday on Monday schedule)	5
7	10/20	The Command Pattern	6
8	10/27	The Adapter and Façade Patterns	7
9	11/03	The Template Method Pattern	8
10	11/10	The Iterator and Composite Patterns	9
11	11/17	The State Pattern	10
12	11/24	The Proxy Pattern	11
13	12/01	Compound Patterns	12
14	12/08	Pattern Categories; and etc.	13
	12/??	**Final Exam**	