LIS452LE
Summer 2016
Instructions for Student Project

Overview

The purpose of the student project in this course is different than that of student projects in more typical graduate courses. Usually, we want the project to be fully conceived, fully executed, and fully delivered before the end date of the course. As such, it represents your mastery of the course concepts, tools, and techniques and it wraps the course up with a nice pretty bow.

By contrast, my goal for your student project in this course is that it be well conceived, and only partially executed and delivered by the end of our course. There are two reasons for this different approach. The first is that it is difficult to go through all of the traditional project steps while you are also learning the Java programming language. The second reason is that I am looking for a project that will keep you involved in the programming world as you work over a period of weeks, months, and perhaps years to finish it. If your project is well chosen, you should see it as the beginning of a hobby or the beginning of a scholarly research project rather than the end of this course.

Subject

The subject of your project can be any product or series of products that can be implemented using the Java programming language. When your project reaches its conclusion, it may be a game, a text mining project, a data repository project, a data analysis project, or any of a thousand other projects that could be implemented using Java. My most important requirement for the subject that you choose is that it be one that interests you enough that you are likely to continue it as a hobby, or as part of your academic pursuits after our course is finished. My motivation here is simple. Students in both higher education and in industry report to me that their programming skills atrophy if they are left unused for a long period of time. This project should be one that will motivate you to open it up every few weeks or months to keep your skills sharp and your knowledge expanding.

The Need for Original Work

The project that you choose and the work that you demonstrate at the end of the class should be yours. Yet, the spirit of Java programming and the spirit of object-oriented programming in general is that we should often design and build classes that can be shared with others. So, to the extent that your project includes the use of classes created by others, you will have demonstrated your understanding of the power of sharing code. If you build a project that includes pieces that can be shared with others (perhaps as an open source project), then you have demonstrated the understanding of the object-oriented software ecosystem even further.

You should also feel free to look at the work of others to get inspiration. This includes inspiration for the general topic of your project as well as inspiration for the details of how to solve a particular programming problem. We have long passed the time when the best programmers are expected to dream up ideas for projects and algorithms for solving programming problems from scratch. In fact, the most highly productive designers and programmers are probably some of the best at building upon the work of others.

Additionally, you should feel free to consult with your colleagues (and me) regarding your project choice and your programming problems. Successful people in this field are highly collaborative. And, more than 20 minutes spent being stuck on the same problem without consulting someone else for their insight and/or help is usually time wasted.

Having said that, please make sure that you identify clearly which work is yours and which pieces have been borrowed from others. Also, please don't copy code from any source without understanding what it does and why it works. You should understand the work that you turn in and you should have done most of the thinking and all of the typing yourself.

The Need to Demonstrate Java Object-Oriented Programming

Most Java courses begin by teaching the procedural features of the Java programming language. Having achieved that, these courses continue by teaching object-oriented concepts and the object-oriented features of the Java programming language. Because courses are often too short, the object-oriented aspects of Java typically get much less attention than they deserve. We, too, will be covering procedural Java before we cover object-oriented Java. Fortunately, we will have the opportunity to cover object-oriented concepts and features in greater depth than is typical. Roughly half of the class will cover object-oriented Java.

With this in mind, I have included the requirement that your project code demonstrate the object-oriented features of Java. If you choose a project that will make use of an open-source Java library, this requirement will be easily met. In using the library, you will be creating object-oriented Java code.

By contrast, you may choose a project that will consist entirely of your own code. In this case, you will need to make a special effort during the design of your project to include object-oriented features in your Java code. This will mean designing one or more custom Java classes that will become part of your solution. Given the amount of attention we will be paying to object-oriented Java in class, this requirement should be easily met.

The Need to Demonstrate Effective Testing Practices

The purpose of testing is to discover errors. While well-tested code is likely to still contain some latent defects, a good testing process should lead to a reliable product by identifying a high proportion of errors early so that they can be repaired.

We will be spending a considerable amount of our class time discussing test planning, testing, and test frameworks. When testing our object-oriented Java classes, we will be using the JUnit testing framework. The JUnit testing framework has become the de facto standard testing tool for Java programmers.

The testing-related requirement for your project is that you have an articulated test plan and that you have test code that you can run that demonstrates your progress in implementing the test plan. In many cases, students will be able to take advantage of the JUnit framework to test their projects. In other cases, use of JUnit may not be practical. So, the use of the JUnit framework in your project is not a specific requirement.

How Much Code Do I Need to Create During the Course?
By the time the project is due on the last day of the course, you should be able to clearly articulate your current vision for the project. That means that you should have a coherent description that you can share with the class. The fact that this vision may change later as you work your way through the project in the coming weeks, months, and years is not a problem. This kind of change is an expected part of continuing discovery.

In addition to being able to articulate the concept for the project, you need to have created at least one non-trivial Java program that works, and that demonstrates that you have made a practical start. How much actual code you have written and tested will depend on how much of your project goal is going to be reached with code that you write. For instance, if you have decided to use open source Java code like the Twitter Java toolkit, it may be sufficient to show that you have identified the toolkit, downloaded a current copy, and written one or two simple test programs that demonstrate that you are on your way to mastering the toolkit.

On the other hand, if you have decided that your project will be a blackjack program that you are going to write from scratch, then you probably will have created and tested more of your own Java code. I will be judging the completeness of your work based on the total effort demonstrated and the extent to which its current state represents a reasonable project start.

As noted above, in addition to delivering the first installment of the “production” code for your project, you also will be expected to deliver the first installment of the “test” code for your project.

What Else Do I Need to Deliver?
You should also have developed **a plan for your next three steps on your project**. They need not be gigantic steps. They need not all be coding steps – some can be research oriented. I want to see that you have a reasonable and concrete plan for how you will proceed when the class is over.

What About the Project Presentation?
On the last day of the class, each of you will make a presentation that takes at approximately 5 minutes. During your presentation, you should expect to be made a presenter on the Blackboard Collaborate platform. From here, you will make your presentation and demonstrate your code. Please create a few presentation slides using PowerPoint or some other equivalent that you can use to keep on track and keep the class informed. My experience is that it is not possible to present more than one slide per minute. So, you will only need a handful of slides.

As for content, I want you to cover the following:

1. The vision for your project
2. Your production code created thus far
3. Your test code created thus far
4. Specific plans for the next 3 steps of your project

Is There Anything Else to Hand In?

Yes. In addition to the slides from your presentation, I am expecting you to turn in an Eclipse project that contains your code in the state in which you wish to have it evaluated. I will expect you to submit this to your Git repository.