CSC 211 - Introductory Programming and Design

#### Laboratory Assignment 01: Introduction to Eclipse

##### Thursday, January 27, 2005

**Due Date:** Thursday, February 3rd, at the beginning of lab.

1. **About this assignment**
   * Don't Panic  
     This assignment may look a bit thick, but that is because most of the lab consists of figures intended to help you create and run an Eclipse project file. The lab itself shouldn't be too long. Still, you may not be able to finish it during the lab time. That is fine, you are not expected to. Lab time is for familairizing yourself with the assignment and asking questions. The assignment is not due until next week.
   * Objectives of the assignment  
     The objectives of this assignment are for you to:
     + Learn how to create a run a Java project using Eclipse
     + Write a working Java program
     + Practice using classes
     + Get familar with how labs will be organized for the rest of the semester
     + Get early feedback on your code formatting style

One thing you should learn early on is that you need to read the assignment carefully before rushing to type code. Read the assignment through once. If you don't understand something, keep going, it may be explained later on. Once you have read the entire assignment, reread it. This time you can stop to ask questions, begin coding your solution, etc.

1. **Create a new Eclipse project**  
   It is important that you learn how to create and configure an Eclipse project because we will need to do so very often during the course of the semester. We will go over each operation step by step together in this lab, however, this section should give you enough information to repeat the process at hom, and in lab next week.  
   Launch the Eclipse program. The program will ask you to select a workspace where your projects will be saved. If you are working in the Envision Lab, you will want to save your files to the Z: drive or removable storage (a zip disk, USB drive, etc.).  
     
   Select New and Project from the file menu or click the appropriate icon on the toolbar and select project.  
     
   Once you have selected Project, you should see a screen with a list of project choices. Select **Java Project** and click the Next > button.  
     
   On the next window, enter the name of your project in the "Project Name" field (lab\_01, for example). The project will automatically be created in the workspace that you chose when you first opened Eclipse. If you would like to change the location, choose Create project at external location. Clicking the Next > button will allow certain default options to be overridden and additional options specified. For now, you can just use the Finish button.  
     
   Eclipse associates Java projects with a Java perspectove, a set of views with information relevant to Java development. When you first create a Java project, you may be asked if the program should switch to the Java perspective. Select yes.  
     
   You should now see the Java workbench where you will work on your project. On the left of the screen you should see the package explorer. Double clicking a class or method in the explorer will bring up the source code in the center of the workbench. The bottom of the screen contains tabs for errors, Javadoc, and the console.  
     
   To create a new file, you need to create a new class. Select New and Class from the File menu.  
     
   The create class wizard allows many features of the class to be specified before the class is created. For now, name the class (for example, Lab) and make sure the box for public static void main (String[] args) is checked.  
     
   You should now have a source file in the center of the workbench with the basic outline of the main class. You can begin editing the file.  
   
2. **Running a Program in Eclipse**  
   Once you have written a program, you can run it in Eclipse by selecting Run As from the run menu, or by clicking on the arrow next to the run button on the toolbar and selecting run as. Most programs we will be writing in this class will be Java applications. You only need to select run as the first time you run the program, after that you can just select run.  
     
   If your program has errors, you will see a message box that says, "Errors in programs, continue launch?". When you return to the workbench, each line with an error should have a red circle with an X in it next to it. Select the Problems tab at the bottom of the screen to see a list of the errors in the program. Once you have fixed the errors, you can try running the program again.  
   
3. **Writing Code**  
   Begin the following project in lab and complete it for homework. All the information you need to complete this assignment can be found in Chapter 2 of your textbook.  
   **Step One:** Create a window that displays the message "Hello and welcome to CSC 211 Lab 1"  
     
   **Step Two:** Create a second window that asks the user for his or her name in the format:  
       first middle last  
     
   **Step Three:** Create a third window that prints out the users first name and asks them to enter their date of birth in the format:  
       MM DD YYYY (for example: 09 06 2000)  
     
   **Step Four:** Create a fourth window that prints out the users full name and the day and date on which they were born.  
   To complete Step 4, you will need to convert a String object to a Date object. You will ask the user for the date in the form MM dd yyyy (for example, "09 06 2000"). You must then convert this string to a string of the form yyyy-MM-dd (for example, "2000-09-06"). Once you have created your new string, you can use the valueOf method to convert it to a Date object. The valueOf method is a method of the Date class which is part of the java.sql package (this is different from the Date object we have been using, which is located in the java.util package). To differentiate between the two classes, in the following example the fully qualified names are used for the classes.  
   java.util.Date bdate = java.sql.Date.valueOf("1776-07-04");  
   java.util.Date bdate creates a new Date object and sets it equal to the value returned by valueOf. java.sql.Date.valueOf() takes a string as an argument and returns a Date object.  
     
   *A Note on Comments*  
   Make sure to add comments to your program. Your textbook discusses comments on pages 41 - 45.  
   Basically, comments should explain in easy to understand language what your code does. Any one should be able to read the comments in your code and get a pretty good idea of what your program should be doing. The reader can then look at the code to understand the small details, but the comments should give them the general idea. You don't need a comment on every line of code, but you also shouldn't use just one comment to explain a large block of code. We will talk more about good commenting as the semester goes.