CMPSC 111
Introduction to Computer Science I
Fall 2016

Practical 3
September 16, 2016
Due in Bitbucket by midnight of the day of your practical
"Checkmark" grade

Summary

In this assignment, you will "seed" one of your past programs with defects, share this defective program with a partner (who will also share a program with you), and then you both will see if you can find the mistakes that were purposefully introduced. This assignment will allow you to further practice the skills of debugging programs and orally expressing your intentions for a program.

Review the Textbook

In addition to studying the slides for Chapters 1 and 2, be sure to read Sections 2.1 through 2.6 of your book as it will give you a good review of the content that we have studied so far. Also, you should briefly review all of the Java programs that you have already written for this class — when you are looking for this program, make sure that you look into both the "share" repository for this course and the "student" repository that you used to submit your assignments. Please see the course instructor if you have questions about topics like variables, expressions, and operators.

Seed Defects into Your Program

Before you start creating the Java program required by this assignment, you should separately type the commands "`cd cs111F2016`", "`cd cs111F2016-<your user name>`", and "`cd practicals`" in your terminal window. Once you are in the `practicals/` directory of your Git repository, you can type the command "`mkdir practical03`" to create a new directory for this assignment. Later, you can run the `gvim` command from this directory when you are ready to begin to modify the required Java program. Please see the instructor if you have problems with these preparatory steps.

Once you have found a program that you have already implemented for this class, review its source code to ensure that you can remember your motivation for programming it the way that you did. Next, copy this program into the `practical03/` directory and then take time to fully comment the code so that it explains all of your intentions. At this stage of the assignment, you can practice using the two commenting standards that are available in the Java programming language. Please use the `javac` and `git` command regularly so that, as soon as you add new comments to the program, you try to re-compile it and, if that works correctly, you commit the new version to your repository. Please see the instructor if you are not sure how to work incrementally in this fashion.

Now you are invited to "seed" five different types of defects into your chosen Java program. As you are adding these defects, please make sure that your partner cannot see the mistakes that you are making. You should aim to insert two types of "bugs" into your program, with the goal of having five defects in total. At least three or four of your defects should be ones that will be detected by the Java compiler. The other one or two defects should be "logic mistakes" that will be accepted by the Java compiler and yet lead to a program that does not produce the type of output that you describe in your program's comments. Please try to add defects that are similar to the mistakes that you have been making during our class and laboratory assignments. While the mistakes should not be overly obvious, they should also reflect the types of errors commonly made by programmers in Computer Science 111. You should take notes as to where you placed the mistakes, taking care to ensure that you partner does not see these notes.

Once you and your partner are done inserting defects into your programs, you should explain to each other the overarching purpose of the programs that you implemented. For instance, if your program produces a diagram in the terminal window, then you should explain what it should look like and why you decided to implement it the fashion that you did. Or, if your program serves an intended purpose (e.g., a menu system for a tip calculator), then you should detail it and highlight your goals for implementing the entire system. As you are discussing these matters with your partner, please be as detailed and specific as is possible, expressing all of the relevant aspects of your creation. Of course, you should be careful to not reveal the location of your program's defects!

Finding and Fixing Seeded Defects

After you have inserted the defects into your program and you have had a thorough conversation with your partner, you should use email or Slack to share the source code of your program with each other. Please note that students should only use email or Slack for code sharing and not, under any circumstances, give their partner access to their entire Bitbucket repository. Once you have received a copy of your partner's file, please make sure that you save it in the `practical03/` directory of your Git repository. Now, try to find and fix all of the defects that your partner inserted into the program. Once you locate and resolve one of these issues, please put a comment into the code to explain the problem that you found and the way in which you decided to handle it. As you fix the "bugs", you should use `git` to transfer the improved version of the program to Bitbucket. Ultimately, your `practical03/` directory in your Git repository should contain the source code for your defective program and the corrected source code for your partner's program. Finally, you should use `gvim` to create a one to three paragraph file, called `response`, that articulates and reflects on your experiences in seeding and finding defects in Java programs.

General Guidelines for Practical Sessions

As you are typing your program in the `gvim` text editor, you should regularly save your files. Once you have created a preliminary version of your program and it compiles and runs as anticipated, you should use the "`git add`" command to "stage" it in your Git repository. Next, you can use the "`git commit`" command to save it in your local repository with a version control message. Finally, you can run "`git push`" to transfer your file to the Bitbucket servers. For this practical assignment, you do *not* have to hand in a hard copy of anything—just upload your Java programs and the "response" document to Bitbucket by using the appropriate `git` commands. Please review your "Git Cheatsheet" and talk with a member of the class, the course instructor, or a teaching assistant if you do not understand how to use some aspect of the Git version control system. Don't forget that it is extremely important for you to keep your Bitbucket repository well organized. Also, you are responsible for ensuring that your repository is shared with the course instructor.

Since this is one of our first practical assignments and you are still learning how to use the Java programming language, don't become frustrated if you make a mistake. Instead, use your mistakes as an opportunity for learning about the needed knowledge and skills in computer science.