

CSE 1321L: Programming and Problem Solving I Lab

Lab 3

Type Systems and Expressions

What students will learn:

- Working with Repl.it
- Printing to the screen (i.e. prompting the user)
- Creating variables and assigning values variables
- Reading input from the user and storing it into a variable
- Doing basic calculations with variables to generate a solution

Overview: In this lab, you're going to continue practicing your coding skills by writing programs that interact with the user and do calculations using variables. The labs below also reinforce the concept of creating variables that hold "intermediate solutions" to avoid having one "giant" equation. What you should focus on is trying to understand the problem, understanding the steps needed to solve it, and then converting them into a working program.

For this lab, you will also be required to use <http://repl.it> . You may ask why (which is totally understandable). Your lab final exam (at the end of the semester) will use Repl.it, so we wanted to make sure you were comfortable with it ahead of time. We also recommend that you use Repl.it for coding up quick solutions – such as following along in lecture. Repl.it is free. It doesn't require a log in, but if you create an account, it will save whatever work you've done – which is a good backup plan. If you choose not to create an account, make sure you backup your work by saving each lab solution below in something like Notepad or TextEdit. We recommend creating an account, but it's your call.

Because you'll be using submitting to the Gradescope autograder, make sure that class names (for the Java/C# folks) are called Lab3A, Lab3B and Lab3C, and file names (for everyone) are called Lab3A, Lab3B and Lab3C (with an extension of .java, .cpp, or .cs). Also, please note that the output of your program needs to match the samples provided.

Lab3A: Credit Cards. Financial advisors will almost always tell you that you should pay for things in cash and avoid credit card debt. Further, they tell you that you should have a small emergency fund that you keep stocked for emergencies like flat tires, dead refrigerators and so on. However, life doesn't always work that way and sometimes we need to charge things. So, for this part of the lab, we're going to write a calculator that calculates your minimum monthly payment on your card.

To create this calculator, we will ask the user for 1) the current balance on their credit card and 2) the APR (Annual Percentage Rate) of the card. To calculate the minimum payment, you will multiply the current balance on the credit card (also called the amount owed) times APR and divide this number by 12 since there are 12 months in a year. The formula is $Amount\ Owed \times APR \div 12 = Minimum\ Payment$. The input APR is a percent, so make sure the APR is in decimal form when you are doing the calculations. The Monthly Percentage Rate is calculated by taking the decimal form of the APR and dividing it by 12 since there are 12 months in a year.

Below is an example run. The user input is in **bold**.

Sample run 1: Amount owed: \$ 2000 APR: 19.75 Monthly percentage rate: 1.646 Minimum payment: \$32.92	Sample run 2: Amount owed: \$ 8500 APR: 29 Monthly percentage rate: 2.417 Minimum payment: \$205.42
---	---

Lab3B: GPA calculator. We're getting more practice making a calculator! GPA is important. It's one of the many things employers look at when recruiting new candidates. You also need a GPA of at least 2.0 to graduate from KSU. GPA is measured by "quality points" using the following scale:

- A = 4 quality points
- B = 3 quality points
- C = 2 quality points
- D = 1 quality point
- F = 0 quality points

Each course counts for a certain number of credit hours. For instance, most courses are 3 credit hours. This lab is a 1 credit hour course. Calculus counts 4 credit hours. To calculate the quality points for one course, multiply the number of hours of that course times the quality points you earn for that course.

To calculate your GPA for the whole semester, you take the total number of quality points earned that semester and divide it by the total number of hours taken that semester.

For this lab, write a program that reads from the user the number of hours and quality points earned for four courses then calculates the total hours, total quality points and GPA. **You should only use floats as the data type for your variables.** An example run is shown below. The user input is in **bold**.

<pre>Sample run 1: Course 1 hours: 4 Grade for course 1: 4 Course 2 hours: 3 Grade for course 2: 3 Course 3 hours: 3 Grade for course 3: 4 Course 4 hours: 4 Grade for course 4: 4 Total hours is: 14 Total quality points is: 53 Your GPA for this semester is 3.78571</pre>	<pre>Sample run 2: Course 1 hours: 4 Grade for course 1: 1 Course 2 hours: 1 Grade for course 2: 4 Course 3 hours: 3 Grade for course 3: 4 Course 4 hours: 3 Grade for course 4: 3 Total hours is: 11 Total quality points is: 29 Your GPA for this semester is 2.63636</pre>
---	---

Lab3C: Coins. We are going to design a program that determines the values of coins in a jar. The program will prompt the user to enter the number coins (i.e. quarters, dimes, nickels, and pennies). It will then print out the number of coins entered for each coin type on separate lines and calculates the total amount of money in the jar. An example run is shown below. The user input is in **bold**.

<pre>Sample run 1: Enter the number of quarters: 2 Enter the number of dimes: 2 Enter the number of nickels: 2 Enter the number of pennies: 2 You entered 2 quarters. You entered 2 dimes. You entered 2 nickels. You entered 2 pennies. Your total is 0 dollars and 82 cents.</pre>	<pre>Sample run 3: Enter the number of quarters: 2 Enter the number of dimes: 3 Enter the number of nickels: 4 Enter the number of pennies: 5 You entered 2 quarters. You entered 3 dimes. You entered 4 nickels. You entered 5 pennies. Your total is 1 dollars and 5 cents.</pre>
---	--

Instructions:

- Programs must be working correctly.
- Programs must be saved in files with the correct file name.
- If working in Java or C#, class names must be correct.
- Programs must be completed and checked before working the assignment.
- Programs must be checked by the end of the designated lab session.
- Programs (source code files) must be uploaded to Gradescope by due date.