

# 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 do that, you'll ask the user for 1) the amount of money they have on their credit card and 2) the "APR" – or Annual Percentage Rate – of the card. We'll keep the calculations simple (so these aren't perfect), but to calculate the minimum payment, you multiply the amount you owe times APR/12 – since there are 12 months in a year. Note: paying just the minimum should be a last resort, since it usually does not make progress towards paying off the card. Note: the \$ below is part of the prompt, not what the user enters. Your program should behave User input is in **bold**.

#### Sample run 1:

Amount owed: \$**2000**  
APR: **19.75**  
Monthly percentage rate: 1.64583  
Minimum payment: \$32.9167

#### Sample run 2:

Amount owed: \$**8500**  
APR: **29**  
Monthly percentage rate: 2.41667  
Minimum payment: \$205.417

**Lab3B: GPA calculator.** Yes! Another calculator! GPA is important. Not only do you need to have a 2.0 GPA to graduate from KSU, it's also one of the many things that employers look in the interview process. If you don't know how that's calculated, you need to understand that it's based on "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

Second, each course counts for a certain number of hours. For example, most courses are 3 hours. This lab is a 1-hour course. If you take Calculus, that's 4 hours. To calculate the quality points that you earn for just one course, multiply the number of hours of that course \* the quality points you earn for that course.

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

For this part of the lab, write a program that asks for the number of hours and quality points earned for four courses, then displays the total hours, total quality points, and GPA. For this part of the lab, **you should only use floats** as the data type for your variables. User input is in **bold**.

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
```

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
```

**Lab3C:** Design and implement a program that determines the values of coins in a jar. The program prints out the total dollars and cents in the jar. The program prompts the user to enter the number of coins (quarters, dimes, nickels, and pennies). Print out the number of coins entered for each coin type on separate lines followed by the total amount of money in the jar as dollars and cents as shown below.

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.
```

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.
```

### **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.