

Computer Science 3-4 Problem Set #1

Welcome to the first problem set! In general, we'll talk about a few extra things when I present the material in class, but in general the packets will have reminders of the important info.

We have only looked at a few basic commands in the C++ programming language so far but there is still plenty we can do with what we have. As a reminder; we made a simple program that added two numbers where we: declared variables, used the cout/cin commands to display information and get information from the user, and stored our calculation in another variable. Here is an example of what that code looked like:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7  int firstNumber = 5;
8  int secondNumber = 17;
9  int userNumber = 0;
10 int sum = 0;
11
12 cout << "Welcome to my number adding machine!" << endl;
13 sum = firstNumber + secondNumber;
14 cout << "The sum of " << firstNumber << " and " << secondNumber << " is " << sum << endl;
15 cout << "Please enter a number: ";
16 cin >> userNumber;
17 sum += userNumber;
18 cout << "Adding " << userNumber << " to the total gives us " << sum;
19
20 }
21

```

Escape Character	What it Does	Arithmetic Symbol	What it Does
\n	New Line	+	addition
\t	Horizontal Tab	-	subtraction
\a	Beep	*	multiply
\\ or \?	Print a \ or ?	/	divide
\' or \"	Print ' or "	%	modulus

Here are this week's problems:

- 1) Using the *cout* command and escape characters, write a program that prints a box, an oval (roughly), an up arrow and a diamond to the screen.
- 2) Write a program that asks the user to enter two numbers, stores the two numbers from the user in variables and prints out the sum, product, difference and quotient of the two numbers. Test your division with $9 / 5$...what do you notice?

In the lecture we declared all of our variables to be type *int* which means they are integer values (no decimals), however it is reasonable to think that we might want to use decimals at some point. To do this, we declare our variable to be a *float* instead (a floating point number...which means we can have as many decimal places as we want). In many cases we might care about having a ton of decimal places (we want to be very precise or have a large number), then we can use *double* instead of *float*.

- 3) For this problem, write a program that reads in the radius of a circle as a *float* or *double* and prints the circle's diameter, circumference and area (which will also all be *float/double* type variables). Use the constant value of 3.14159 for π .
- 4) Write a program to evaluate the fuel consumption of a car. Have the user input the miles on the car's odometer at the start and end of the journey, and also the fuel level in the tank (in gallons) at the start and end of the journey. Calculate fuel used, miles travelled, and the overall fuel consumption in miles travelled per gallon of fuel. Print these values, accompanied by appropriate messages indicating what they are.