Note: for homepage blurb use about 22 words followed by “...”

PS1

The first problem sheet consisted of two problems. The first problem was to use one of Newton’s equations of motion to calculate where a cannon ball will land on the floor when it is shot from a cannon. We had to calculate this distance for a range of speeds and angles that the cannon was held at. This data had to be printed out in a table and the table had to look like the given table. In order to do this, we had to use Java’s string formatting.

The second problem in this problem sheet involved creating our first class in Java. The class we had to create was called Cube. As to be expected this had one field, length, and several non-static methods like area and volume. We then had to create a test file to see if these methods returned the results we expected. For example, we expected a cube of length 1 to have a volume of 1.

PS2

For the second problem sheet we were asked to make our first applet. We were given an image of an outline of a house on a coordinate axis. We had to recreate this exactly and then embellish it using Java’s graphics methods.

PS3

Problem sheet three consisted of two problems. Both problems involved working with strings. The first problem was to allow a user to input a word and then program tells the user if the word was a palindrome or not.

The second problem allowed the user to count the occurrences of each letter of the alphabet in a text file. The user chooses a file that they wish to analyze and the program parses all the text from it, counts the letters and outputs a table with the totals for each letter.

Please note that in order to use this program you have to compile it in the command line with “javac Letters.java”. The file you want to analyze also has to be in the same location. If you have done both of these things, “java Letters words.txt” will run the program on the file (if you name it words.txt).

In both of the problems we also had to deal with our first exceptions. For example, if a user input a number when they should have input a word, we didn’t run the program and we told the user that they should try again.

PS4

Problem sheet four had one task, make a hangman game. If you are not familiar with hangman, it involves guessing a secret word. You try to guess each letter at a time and you win if you manage to get all the letters and reveal the secret word, before your number of guesses are up.

This was the first time that we had to produce a program that had several different tasks to perform in order to create a functioning game. In order to do this, we had to break the problem up into sub-problems and organize our code accordingly. For example, if a particular task had to be carried out more than once, we created a function for it. Thereafter, we only had to name this function and use some arguments to carry out said task.

Note that this code needs to be compiled from the command prompt in a similar way to problem sheet three’s Letters program. You will also need a list of words saved in a text file so that the game can select a secret word for you.

Dates:

October 22, 2019

November 15, 2019

December 9, 2019

January 09, 2020