Lab 1 Build Your Own Class

Customer.java is due at the end of your lab. Click the "Submit" button

Objectives:

- Sign up for the lab
- Lab environment: Notepad++, simple DOS commands, compile/run Java programs
- Java review: data types, functions and file input
- Design, implement and test a user defined data type (class)

1. Lab Environment

A card key is not required during the hours when ECS building is open (7:00am-11:00pm Mon. – Fri. except holidays). If you need to work in the lab when the building is closed, you may purchase a card key at the bookstore. The cost is $10.00 (non-refundable).

You are required to have a csc account to access the course website and the computers in the labs. If you don't have one already, activate your csc account from this website: https://connex.csc.uvic.ca/portal/site/itsupport/page/88ff914f-369a-4f5c-9f02-a47c610745ad

There are two drop-in labs with the same environment: ECS 266 and ECS 348.

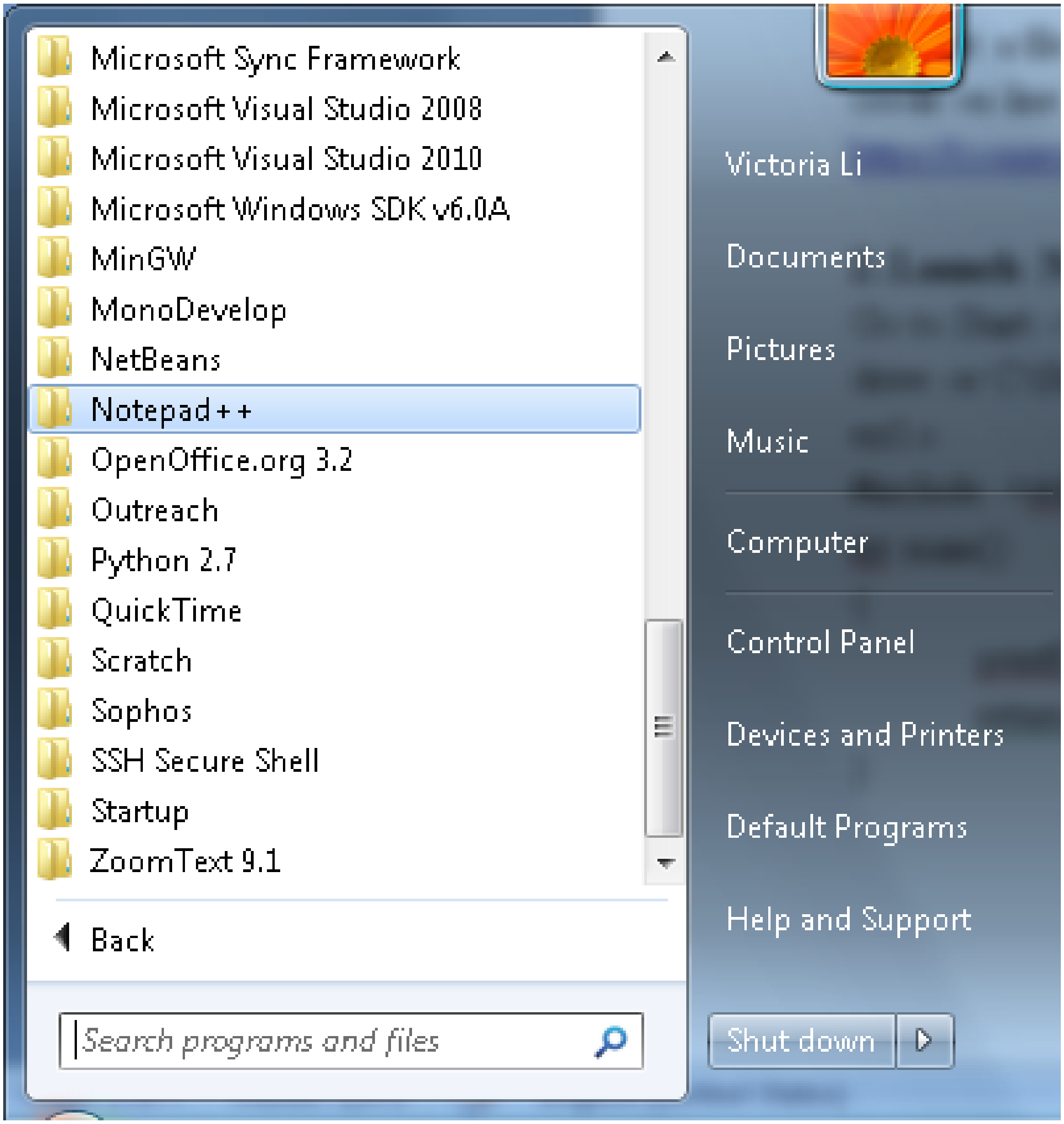
H: drive is your network space in the file server. The data stored in the H drive is persistent and portable: you may access it at any time and from any computers in the labs. "*C:\Users\loginID*" is on your local computer. For example, my loginID is "vli", I can store the Java source file in "*C:\Users\vli*". The advantage to store files locally is that it runs faster, but the disadvantage is that you can't access it from other computers in the labs and it is temporary - the documents stored in "*C:\Users\vli*" could be erased by some system administration scripts overnight.

To print a file from the drop-in labs, you need to have a csc account and purchase some printing credit online at https://www.csc.uvic.ca/PrintPagePurchasing/.

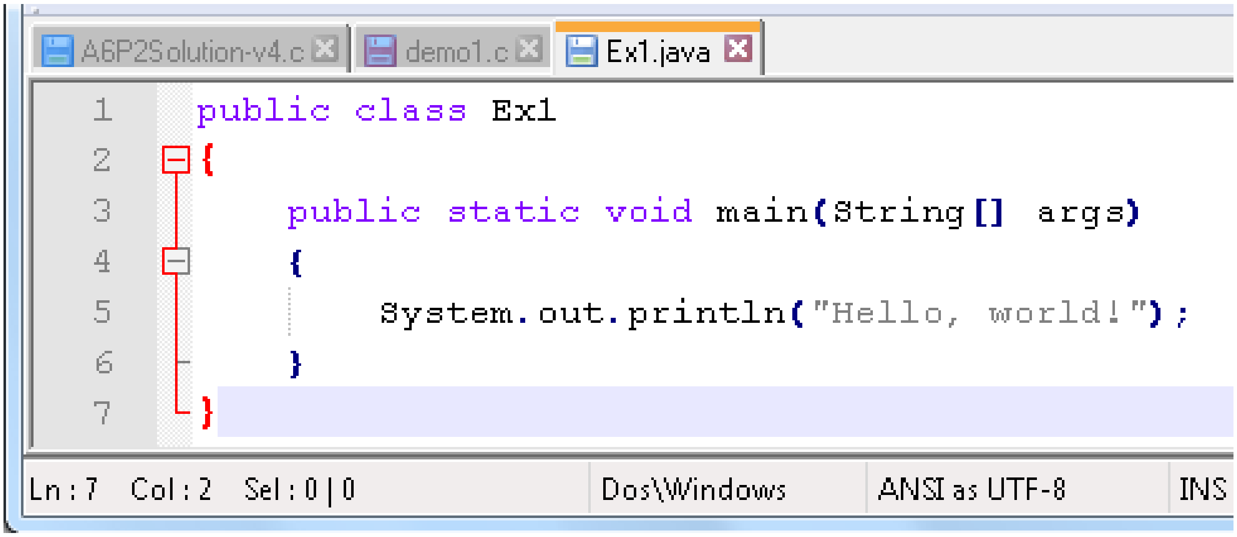
Business Hours for the Assistance Center can be found at: https://connex.csc.uvic.ca/portal/site/cscassist

2. Launch Notepad++ to write a simple Java program

Go to Start -> All Programs -> Notepad++, type the following program and save it as Ex1.java in H drive or "*C:\Users\loginID*"

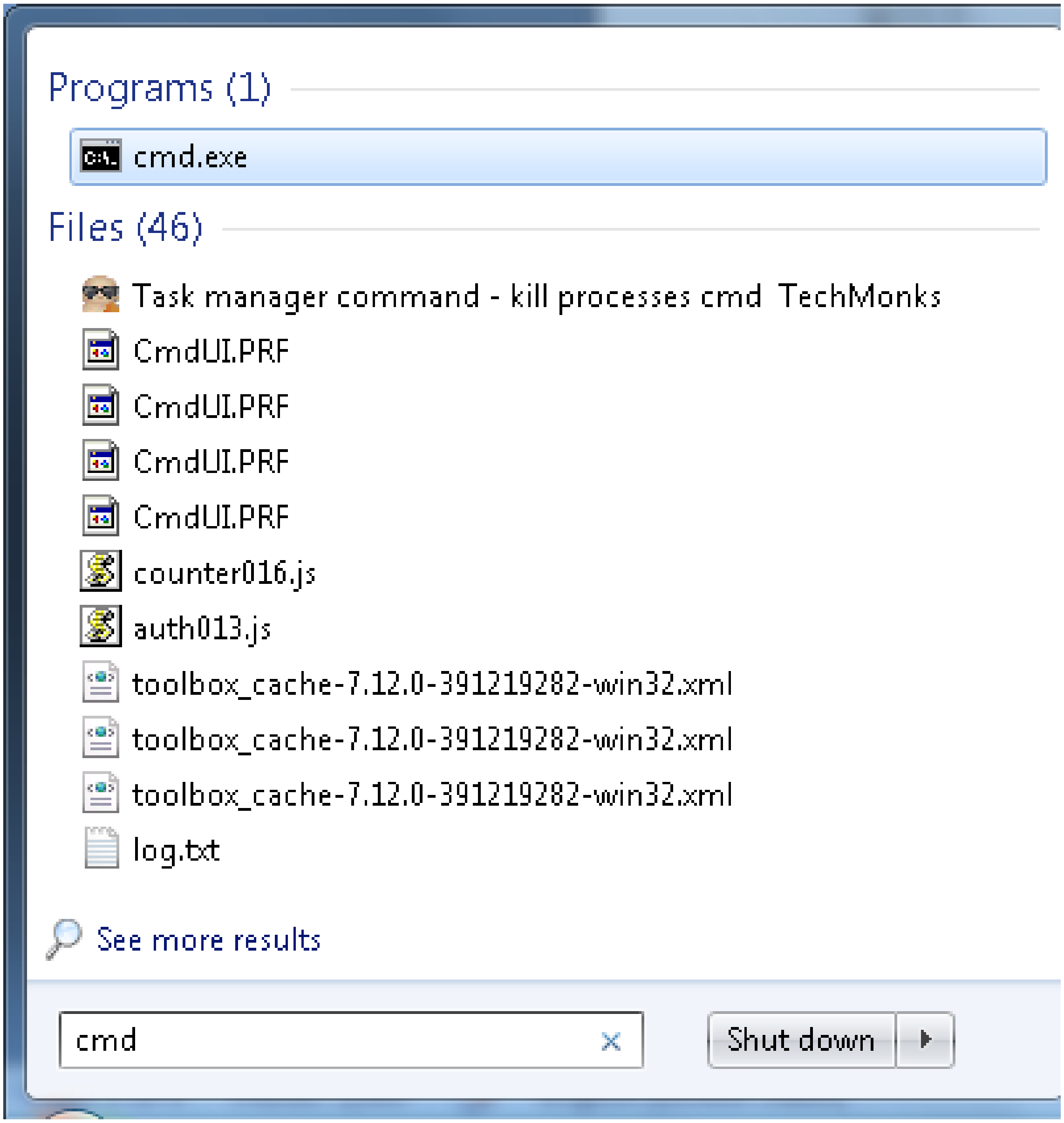


Ex1.java



```
public class Ex1
{
    public static void main(String[] args)
    {
        System.out.println("Hello, world!");
    }
}
```

Compile the program: launch the command window: Start -> Type "cmd", then press enter.



Programs (1)
cmd.exe
Files (46)
Task manager command - kill processes cmd TechMonks
CmdUI.PRF
CmdUI.PRF
CmdUI.PRF
CmdUI.PRF
counter016.js
auth013.js
toolbox_cache-7.12.0-391219282-win32.xml
toolbox_cache-7.12.0-391219282-win32.xml
toolbox_cache-7.12.0-391219282-win32.xml
log.txt
See more results
Shut down

```
cmd
```

At the command window, change the directory to where the Ex1.java is stored. If yours is stored at *C:\Users\loginID*, you don't need to change the directory. It is there by default. If yours is stored at H drive, then type *H*, press enter.
To look at the content of the current directory: type "dir"
To compile the program: type "javac Ex1.java"
If there are no compile errors, type "dir" to see if "Ex1.class" is generated.
Type "java Ex1" to run the file. The output is:
Hello, world!
To find the version of Java installed in the lab: "java –version", you will find that it is

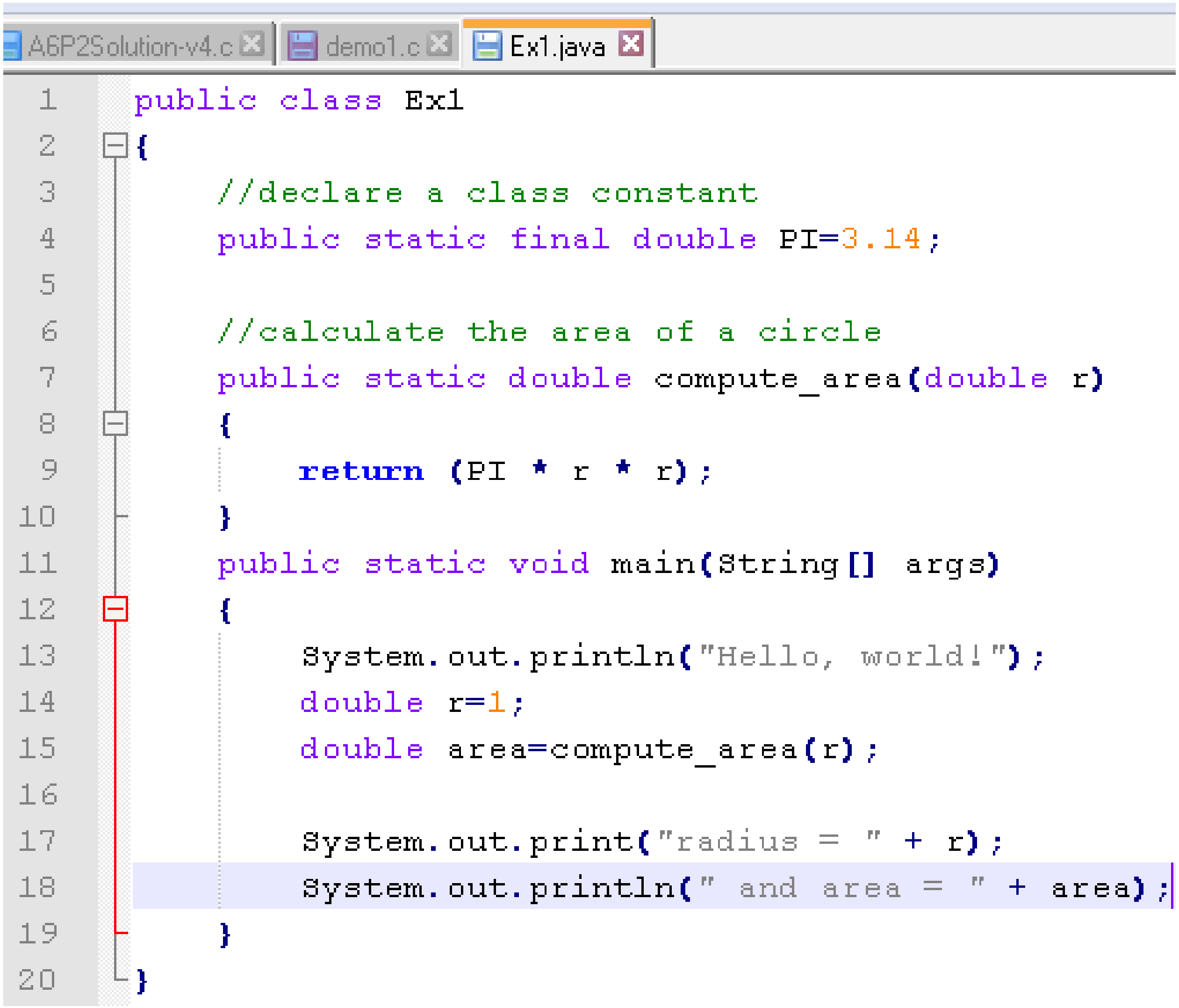
java version "1.8.0_171"

3. More about MS-DOS commands

MS-DOS stands for Microsoft Disk Operating System. It is a non-graphical command line operating system created for IBM compatible computers. Learn more below:

H:	-go to H drive
C:	-go to C drive
.	-current directory
..	-the parent directory
cd ..	-go to the parent directory
dir	-show the files and directories in the current directory.
dir /w	-the same as dir, but use wide list format
ren old new	-rename the *old* file to new *file* name
del file	-remove *file*

4. Java Review - method



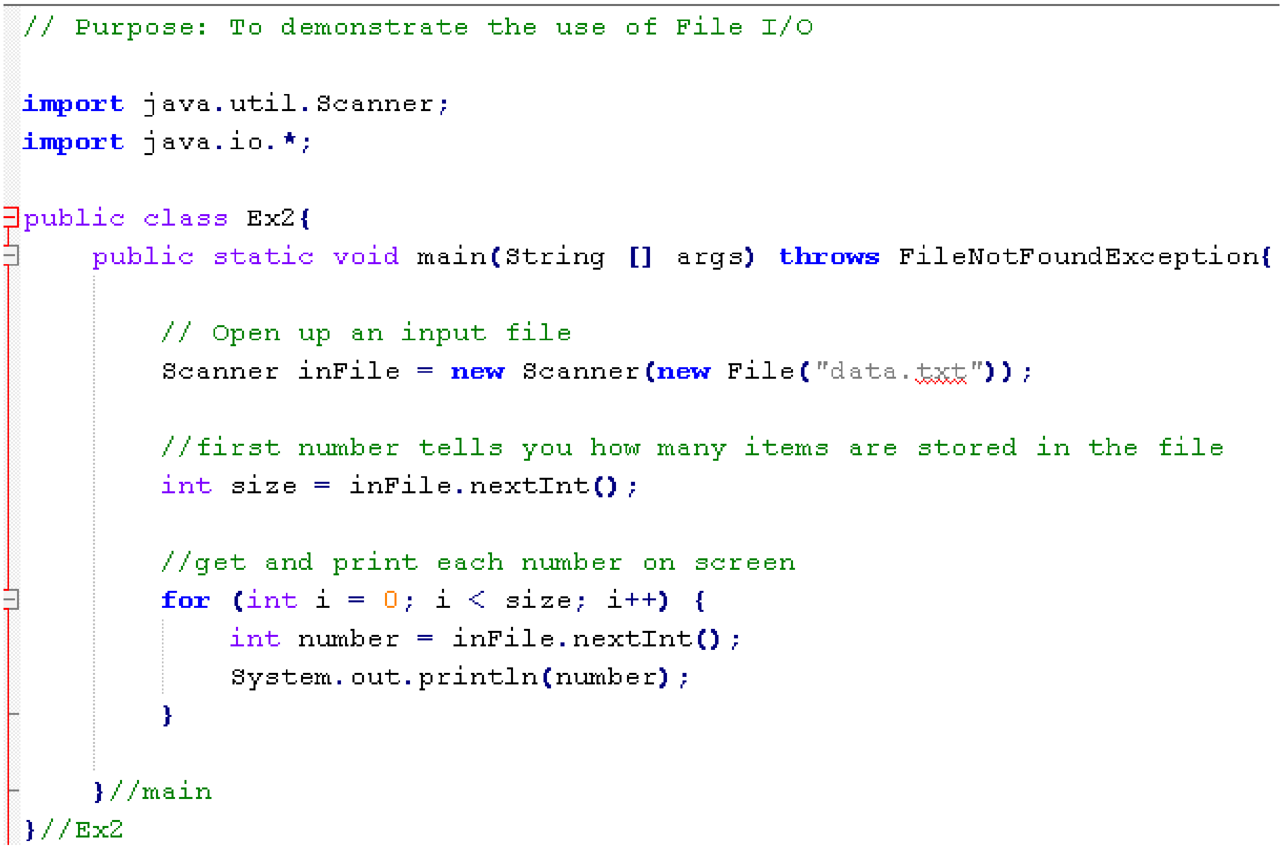
```java
public class Ex1
{
    //declare a class constant
    public static final double PI=3.14;

    //calculate the area of a circle
    public static double compute_area(double r)
    {
        return (PI * r * r);
    }
    public static void main(String[] args)
    {
        System.out.println("Hello, world!");
        double r=1;
        double area=compute_area(r);

        System.out.print("radius = " + r);
        System.out.println(" and area = " + area);
    }
}
```

5. File Input

We usually get input from a file to process bulk data. Here is some sample code. Type the code and run the program. Notice you need to import some standard libraries. Exception is going to be covered later on. You need to download a data.txt file.



```
// Purpose: To demonstrate the use of File I/O

import java.util.Scanner;
import java.io.*;

public class Ex2{
    public static void main(String [] args) throws FileNotFoundException{

        // Open up an input file
        Scanner inFile = new Scanner(new File("data.txt"));

        //first number tells you how many items are stored in the file
        int size = inFile.nextInt();

        //get and print each number on screen
        for (int i = 0; i < size; i++) {
            int number = inFile.nextInt();
            System.out.println(number);
        }

    }//main
}//Ex2
```

6. Design your own Abstract Data Type (ADT)

Let's model a simple real life entity. First, identify the attributes, write them down; then identify the behaviors, write them down. Let's identify the ones relevant to our program, remove the irrelevant ones.

Entity Name:
Attributes:
Behaviors:

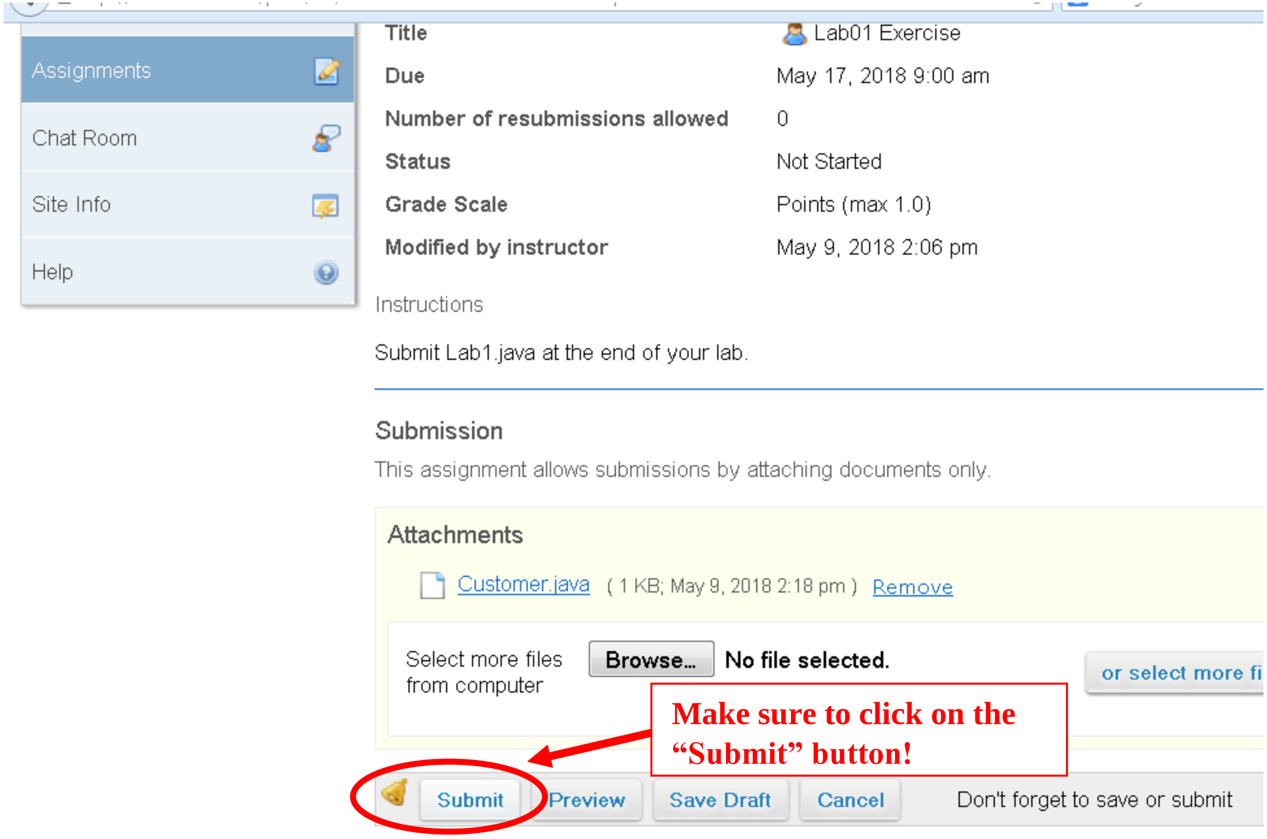
7. Implement your own Abstract Data Type (ADT)

Then implement the ADT above using Java. We need to decide the data types for each attribute, the prototypes of the methods, and decide which ones should be private, which ones should be public. Give a meaning full name for the class. For example, if you are going to model customer for a telephone company, call it Customer.java.

8. Testing

Use stubs in your class, then write a tester. This way, you can start testing your program when you finish implementing the first method.

9. Submit *Customer.java* at the end of your lab.



Title	Lab01 Exercise
Due	May 17, 2018 9:00 am
Number of resubmissions allowed	0
Status	Not Started
Grade Scale	Points (max 1.0)
Modified by instructor	May 9, 2018 2:06 pm

Instructions

Submit Lab1.java at the end of your lab.

Submission

This assignment allows submissions by attaching documents only.

Attachments

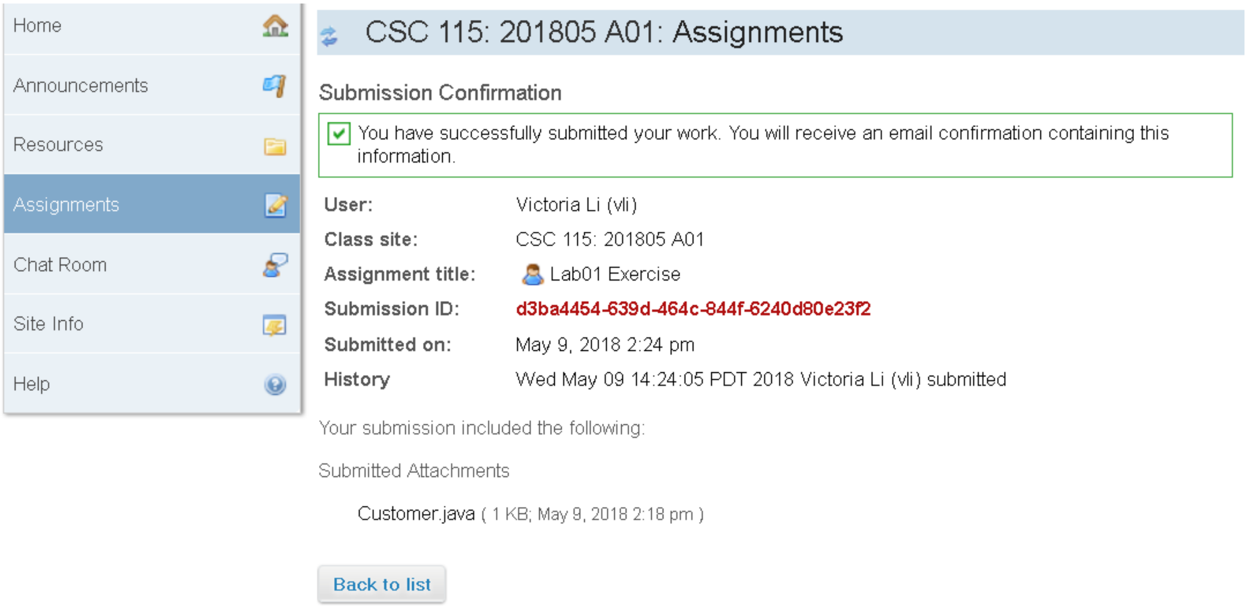
Customer.java (1 KB; May 9, 2018 2:18 pm) Remove

Select more files from computer No file selected.

Make sure to click on the "Submit" button!

Don't forget to save or submit

You will see a screen similar like this. Take a screenshot (press both *Alt* and *Print Screen* keys of the keyboard) and paste it in a word document, save the file as a proof of submission in case you need to appeal your grade later on.



CSC 115: 201805 A01: Assignments

Submission Confirmation

☑ You have successfully submitted your work. You will receive an email confirmation containing this information.

User:	Victoria Li (vli)
Class site:	CSC 115: 201805 A01
Assignment title:	Lab01 Exercise
Submission ID:	d3ba4454-639d-464c-844f-6240d80e23f2
Submitted on:	May 9, 2018 2:24 pm
History	Wed May 09 14:24:05 PDT 2018 Victoria Li (vli) submitted

Your submission included the following:

Submitted Attachments

Customer.java (1 KB; May 9, 2018 2:18 pm)