# Lab 1

**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)
- Submit one file: *Lab1.java* at the end of the lab. Click the "Submit" button

## 1. Lab Environment

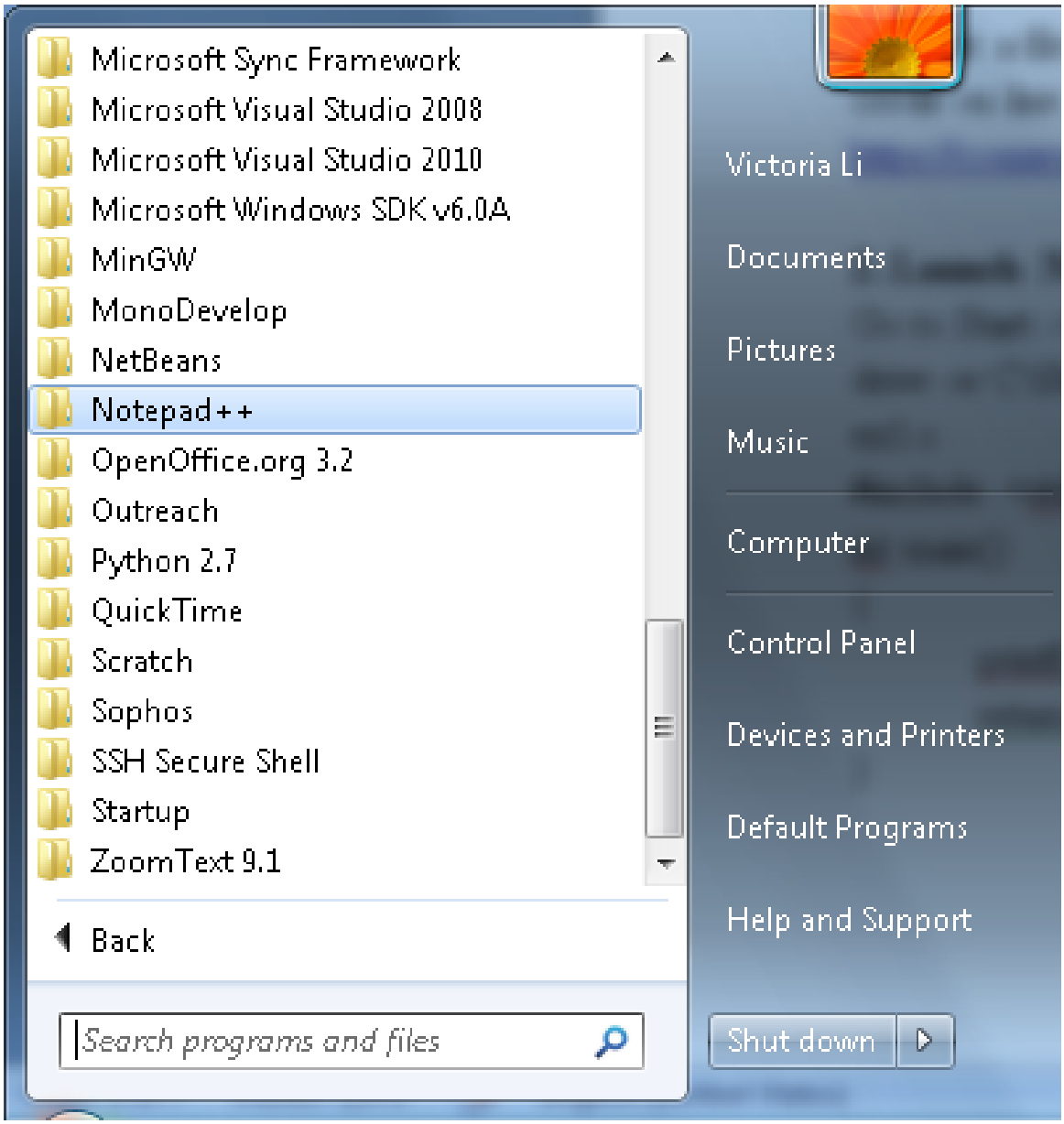
H: drive is your network space in the file server. The data stored in the H drive is persistent and portable: you may access it at any time and from any computers in the labs. "*C:\Users\loginID"* is on your local computer. For example, my loginID is "vli", I can store the Java source file in "*C:\Users\vli"*. 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.

A card key is not required during the hours when ECS building is open (7:30am-10:00pm Mon. – Fri.). 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).

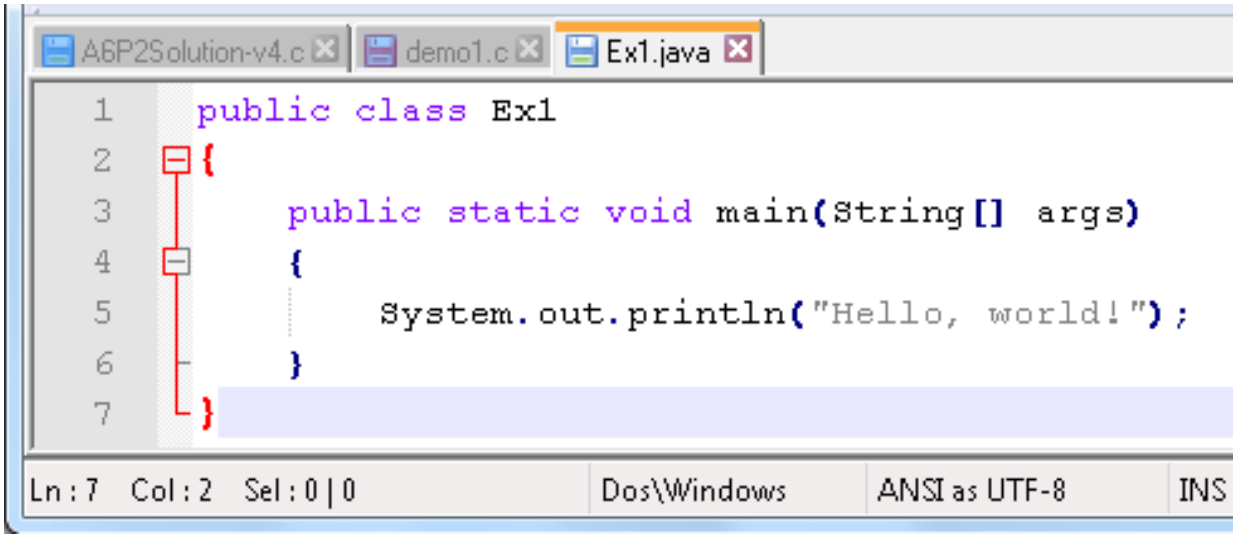
To print a file from the teaching labs, you need to have a csc account and purchase some printing credits online at
https://www.csc.uvic.ca/LPP/index.html

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

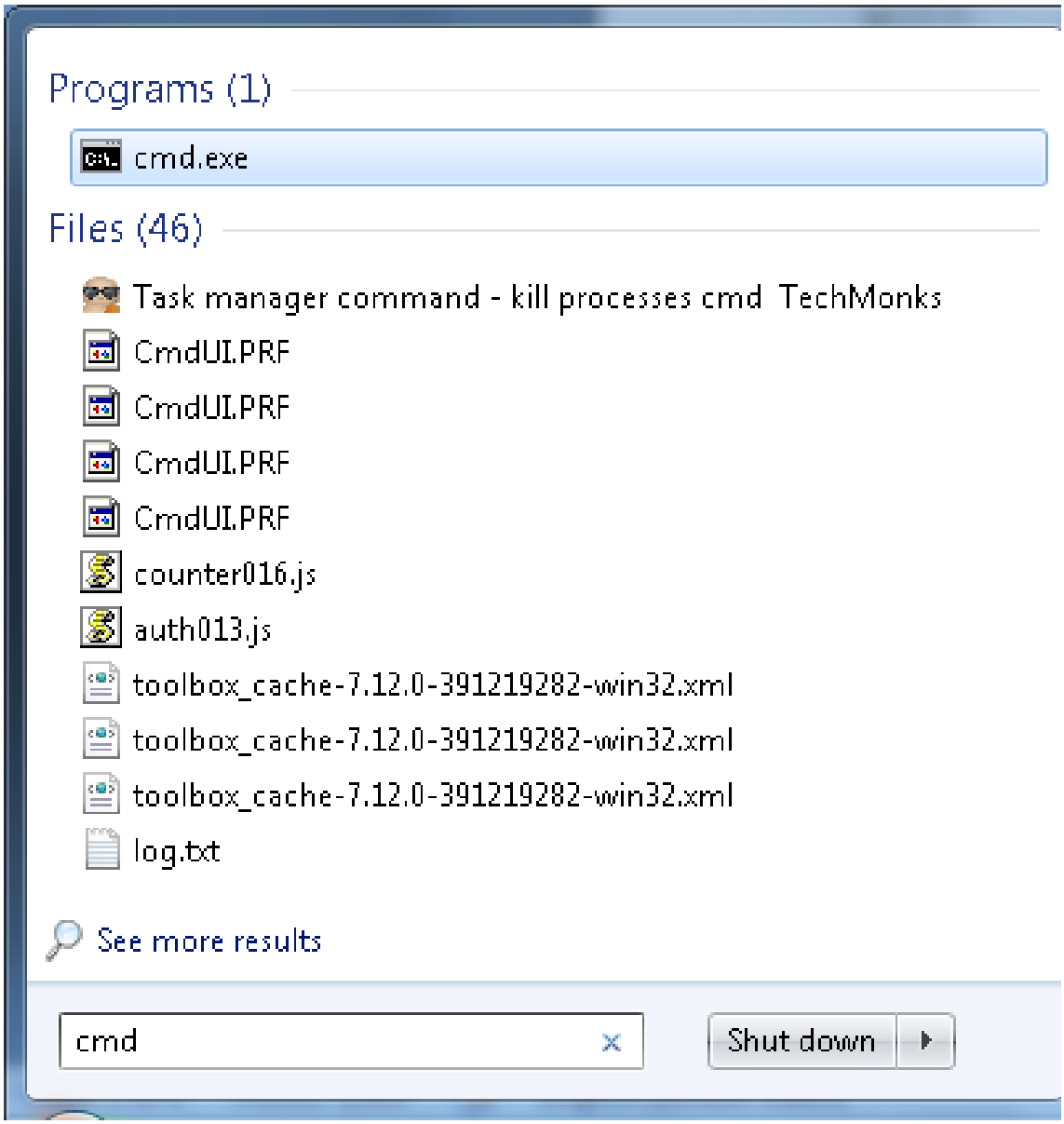
Go to Start -> All Programs -> Notepad++, type the following program and save it as Ex1.java in H drive or "*C:\Users\loginID*"



**Ex1.java**


A6P2Solution-v4.c
demo1.c
Ex1.java
1 public class Ex1
2 {
3 public static void main(String[] args)
4 {
5 System.out.println("Hello, world!");
6 }
7 }
Ln : 7 Col : 2 Sel : 0 | 0
Dos\Windows
ANSI as UTF-8
INS

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




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!

### 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 |
| **.** | -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

A6P2Solution-v4.c | demo1.c | Ex1.java

```
1  public class Ex1
2  {
3      //declare a class constant
4      public static final double PI=3.14;
5
6      //calculate the area of a circle
7      public static double compute_area(double r)
8      {
9          return (PI * r * r);
10     }
11     public static void main(String[] args)
12     {
13         System.out.println("Hello, world!");
14         double r=1;
15         double area=compute_area(r);
16
17         System.out.print("radius = " + r);
18         System.out.println(" and area = " + area);
19     }
20 }
```

**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. Call it Lab1.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.

**Use Connex to submit one file: *Lab1.java* at the end of the lab. Make sure you click the "Submit" button.**