CS 116- ASSIGNMENT TO BE PERFORMED IN CLASS. NO SUBMISSION REQUIRED!

THIS ASSIGNMENT DOES NOT REQUIRE SUBMISSION AND WILL NOT BE GRADED. YOU ARE REQUIRED TO SHOW TO ONE OF THE TAs, OR THE INSTRUCTOR, THAT YOU COMPLETED THE WORK BY THE END OF THE CLASS PERIOD AND THAT YOU CAN USE COMMAND LINE TO COMPILE AND INTERPRET.

THIS EXERCISE IS A REVIEW OF BASIC CONCEPTS FROM CS115 AND LEARNING HOW TO USE COMMAND LINE TO COMPILE AND INTERPRET (IF YOU FORGOT HOW).

PART A

1. Work through the instructions of using command line provided at the end and, also posted on the course’s web site under CompileUsingCommandLine.doc. in the schedules.html page.

You must know how to compile and interpret command line using a DOS window and typing the proper command for the compiler or the interpreter.

PART B

2. Create a Service (Template) class that is intended to track data for the category House. Use Notepad++ to write the source code for the class.

- Give the class a name. Make sure that there is NO main method in this class.

- Come up with at least three attributes of your own that have to do with the concept of House. Decide the attributes based on the type of data that you want to track for

Acknowledgement: Material provided by George Koutsogiannakis (Computer Science, IIT)
the category of houses i.e number of bedrooms, the price of a house, the square footage of a house etc.

- Declare those three instance variables by using the proper data type and an identifier of your own choosing. Initialize the declared identifiers to an initial value of your own choosing. These are the instance variables of the service class!!!

- Create a default constructor that initializes the instance variable of the class to some fixed default values other than the initial ones in the declarations of the instance variables.

- Create a non default constructor that initializes the 3 instance variables to the arguments passed to the constructor. The constructor takes as arguments the 3 data types corresponding to the 3 instance variables. Make sure that you use different identifier names s arguments than the names chosen for the instance variables.

- For each of the instance variables create an accessor method.

- For each of the instance variable create a mutator method.

- Create a toString method that outputs (returns) a String that includes the values of all 3 attributes (of all 3 instance variables).

- Compile the class to get the .class file. NOTE: YOU MUST COMPILER COMMAND LINE!!

3. Now let us create a Client class to the above Service class.

- Give the class a name. Make sure that there is a main method.

- Inside the main method create 2 objects of the Service class; one by calling the default constructor and the other by calling the non default constructor. Call the first object (created with the default constructor) house1.
- Call the second object house2 and give to the non default constructor specific values for the attributes that you have chosen

House house2 = new House(attribute values go here)

- Access the value of the first and second attribute for each object. Output (display) each value by using System.out...

For instance to access the value of an attribute called bedroomNum, which has been declared to be of data type int, for an object called house1 the code will be:

```java
int bn = house1.getBedroomNum();
```

where House1 is the identifier given for a House type object (by the Client class), followed by the dot operator (.), followed by the name of the accessor method representing the specific attribute (bedroomNum in this example). This is called an invocation (and it is similar to the method call concept that we have studied). Since the accessor method returns the data type nd its value for that attribute we need to create a new identifier in the main method to capture the returned value. Therefore we will capture the returned value with the identifier bn of type int in this example.

We can now display the value by using the code:

```java
System.out.println("The number of bedrooms in House house1 is:"+bn);
```

- Modify the value of the first parameter of the first object by calling the proper mutator method and give it a new value. Now, access the value of the first attribute again for the first object and then output the value by using System.out.....

- Add a call (invocation) to the toString method and output on the screen the values of all 3 attributes for object house1 i.e

```java
String output=house1.toString();
System.out.println(output);
```
• Add a call (invocation) to the toString method and output on the screen the values
  of all 3 attributes for object house2 i.e
  String output1=house2.toString();
  System.out.println(output1);

**RUN THE PROGRAM COMMAND LINE!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!**

**TO RUN COMMAND LINE OPEN A DOS WINDOW.**

**SET THE PATH TO WHERE YOUR JAVA FILES ARE LOCATED.**

**TYPE THE JAVA COMMAND TO COMPILE THE FILES I.E.**

C:/javac MyProgram.java

**TYPE THE JAVA COMMAND TO RUN THE PROGRAM I.E.**

C:/java MyProgram

Example output of a program whose service class has attributes:

double price
String suburb
int numOfBedrooms

/*

-------- Java Interpreter --------

The price of h1 is: 100000.0 and the suburb is: Glenco

The price of h2 is: 200000.0 and the suburb is: Glen_Elyn

The new price for h1 is: 300000.0

The information of h1 is: The price is  300000.0 The suburb is Glenco The num of bedr. is 3

The information of h2 is: The price is  200000.0 The suburb is Glen_Elyn The num of bedr. is 4

Output completed (0 sec consumed) - Normal Termination*/
CS 116---COMPILE/INTERPRET USING COMMAND LINE

PURPOSE:

- GO THROUGH THE STEPS OF CREATING A SIMPLE JAVA PROGRAM.
- LEARN HOW TO UPLOAD THE ASSIGNMENT’S ZIPPED FILE ON BLACKBOARD.

STEP 1: Open the text editor program called EditPlus and from the file menu choose FILE->New->Java

STEP 2: The Editor displays the template of a normal Java program. We can modify the template.

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

Let us modify the template by adding additional instructions to the template so that it looks as follows when you are done:

(Note: In typing the program I would like you to follow the indents as shown. Advance the tabs to maintain the position of the curly brackets and the typing of the programming instructions as shown.

Make sure that you type the comments as shown preceded by //)

    public class MyFirstProgram
{ //double slashes are used for comments
  //anything after the double slashes is disregarded by the compiler
  //the name of the class is: MyFirstProgram
  public static void main (String[] args)
  {
    System.out.println("This is a Java program");
  }
}

**STEP 3:** Now, we are ready to save the program in a file. Choose from the EditPlus menu File->SaveAs

**MAKE SURE THAT THE PROPER DIRECTORY IS SHOWN WHERE YOU WANT TO SAVE THE SOURCE CODE FILE.**

IN THE WINDOW THAT HAS APPEARED, RIGHT CLICK THE MOUSE AND CHOOSE NEW (FOLDER). CREATE A NEW FOLDER ON THE DESKTOP WITH THE NAME: PRACTICEEX1

WE WANT TO SAVE THE PROGRAM THAT WE CREATED IN THIS NEW FOLDER. BEFORE YOU CLICK THE SAVE BUTTON YOU NEED TO NAME THE FILE (REFERRED TO AS THE SOURCE CODE FILE) THAT YOU CREATED.


i.e. MyFirstPorgram.java

CLICK O.K.
STEP 4: IN WINDOWS EXPLORER GO TO THE FOLDER C:\PRACTICEEX1 AND VERIFY THAT THE FILE MyFirstProgram.java IS THERE.

STEP 5: WE ARE GOING TO USE THE JAVA COMPILER TO COMPILE THE SOURCE CODE FILE WE CREATED. THERE ARE A COUPLE OF DIFFERENT WAYS WE CAN DO IT.

1. WE COULD DO IT BY OPENING A DOS WINDOW AND TYPING THE PROPER COMMAND. THIS IS CALLED COMMAND LINE. THE OUTPUT WILL APPEAR ON THE DOS WINDOW (SOMETIMES REFERRED TO AS THE DOS PANE).

2. WE COULD DO IT DIRECTLY FROM EDITPLUS BY CHOOSING THE JAVA COMPILER FROM THE TOOLS MENU AT THE TOP TOOLBAR. IN THIS CASE THE OUTPUT WILL APPEAR AT THE BOTTOM OF THE EDITPLUS SCREEN.

WE WILL DO IT BY USING METHOD 1 (COMMAND LINE):

FROM All Programs CHOOSE Assesories->Command Prompt

A BLACK SCREEN WINDOW WILL APPEAR. THE PICTURE BELOW IS A DOS SCREEN.
NOTICE THAT THE SCREEN SHOWS A PATH ON YOUR COMPUTER LIKE:

C:\Users/George> _

OR IT MAY BE A DIFFERENT PATH. WE NEED TO NAVIGATE TO WHERE THE FOLDER WITH OUR
FILE MyFirstProgram.java IS LOCATED. WE CAN MOVE UP OR DOWN THE DIRECTORY PATH BY
USING DOS COMMANDS.

i.e. C\User/George> cd..

- TYPING cd FOLLOWED BY TWO DOTS AND PRESSING Enter KEY, MOVES THE CURSOR TO
  THE LOWER DIRECTORY.
  C\User>

- TYPING THE SAME AGAIN AND PRESSING THE Enter KEY, MOVES THE CURSOR DOWN
  ONE MORE DIRECTORY
C\User>cd..

NOW THE CURSOR IS AT C:\>_

• THE COMMAND `cd` FOLLOWED BY SPACE AND THEN THE NAME OF THE DIRECTORY (FOLDER) THAT WE WANT MOVES THE CURSOR TO THAT DIRECTORY.

C:\>cd PRACTICEEX1

PRESS Enter KEY.

RESULTS IN

C:\PRACTICEEXE1>_ 

• TYPING C:\PRACTICEEXE1>dir

SHOWS ON THE DOS WINDOWS ALL THE FILES IN FOLDER PRACTICEEX1. YOU SHOULD TRY TO VERIFY ONCE AGAIN THAT THE FILE MyFirstProgram.java IS THERE BY TYPING THE DOS COMMAND `dir`.

NOW WE ARE READY TO USE THE DOS COMMAND THAT WILL ALLOW US TO COMPILE OUR SOURCE CODE.

TYPE THE COMMAND AS FOLLOWS:

C:\PRACTICEEXE1>javac MyFirstProgram.java

NOTICE THAT THERE IS SPACE AFTER `javac` (which is the command for the java compiler)

PRESS Enter KEY.

THE FILE MyFirstProgram.class IS CREATED. THIS FILE IS CALLED THE BYTECODES FILE.

• TYPING THE DOS COMMAND `dir` WILL DISPLAY ALL FILES AGAIN. YOU SHOULD BE ABLE TO VERIFY THAT THIS NEW FILE WAS CREATED.

• HERE IS AN EXAMPLE SCREEN THAT SHOWS HOW I GOTTO THE DESKTOP ON MY LAPTOP (the path may be a little different in your computer):
C:\Users\George>cd Desktop

C:\Users\George\Desktop>cd Practice1

C:\Users\George\Desktop\Practice1>

Typing the DOS command dir after the above prompt will display all the file residing in folder Practice1 (that is the file we created and saved there under the name MyFirstProgram.java)

**STEP 6:** Let us now interpret (execute) the program we created. In the same DOS window as before type the command to call the java interpreter:

C:\PRACTICEEX1>java MyFirstProgram

NOTICE THAT THE COMMAND IS java (not javac) THEN SPACE AND THEN THE NAME OF THE BYTECODES FILE **WITHOUT THE EXTENSION .class.**

THE OUTPUT OF THE PROGRAM SHOULD APPEAR ON THE DOS WINDOW.

THE PICTURE BELOW ILLUSTRATES THE PROCESS DESCRIBED. THE DIRECTORY PATHS IN YOUR CASE MAYBE DIFFERENT THAN THE PICTURE.
COMPILER GENERATED ERROR

BECAUSE THE .java EXTENSION WAS NOT TYPED AFTER THE NAME OF THE FILE