# *Programming I (420-B10-HR)*

# *Lab 2 - Introduction to Java and* the Java Development Kit (JDK)

Date assigned: Tuesday, September 1, 2015

Date due: **Tuesday, September 1, 2015**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

1. create a simple Java program in Notepad;
2. compile and execute the program using the Java Development Toolkit (JDK);
3. list the files in a folder in DOS;
4. use the **System.out.print()** and **System.out.println()** methods to display output;
5. declare and initialize integer and real variables;
6. perform arithmetic operations using integer and real variables.

**To be handed in:**

1. The ***username*\_B10\_L02\_JDK** folder should be zipped and uploaded to **Moodle**. ***Username*** isyour logon username.
2. The **Moodle Lab 2 Review Quiz** should be completed.

**Important Note:** Throughout this lab replace ***username*** with your logon username.

**To start:**

1. Create a folder called ***username*\_B10\_L02\_JDK** in your **420-B10\Labs** folder.

# Creating a Java source file in Notepad

**Purpose**: To learn to use Notepad to create a Java source file.

**To Do**:

## From the **Windows** main screen:

### **Start** *->* **All****Programs** *->***Accessories** *->* **Notepad**

## You should have a blank screen with the cursor positioned in the upper left hand corner. Type in the following exactly as shown replacing my name with your name. Be sure to capitalize exactly as shown below. Use the TAB key to do any indenting.

/\*\*

\*

\* <p>Course: 420-B10 Programming I</p>

\* <p>Lab 2: Printing on the screen</p>

\* <p>Description: This program prints the Heritage College address on the screen.</p>

\* @author Sandra Stark

\*/

public class Heritage

{

public static void main(String[] args) //method header

{ // main method

System.out.println("Heritage College");

System.out.println("325 boul. Cite des Jeunes");

System.out.println("Gatineau, QC");

System.out.println("J8Y 6T3");

} // main method

} // Heritage class

## Select **Save** from the **File** menu.

## Select your **420-B10\Labs\*username*\_B10\_L02\_JDK** folder in your **H:** drive from the **Save in:** drop-down box.

## Select **All Files** from the **Save as type:** drop-down box.

## Type **Heritage.java** in the file name box. (Be sure to type it exactly as shown with a capital *H* and lower case *eritage.*)

## Click the **Save** button.

# Compiling a Java Program

***Purpose:*** To compile a Java program using the JDK compiler.

To Do:

## Select **Command Prompt** from **Accessories**to the start the DOS command prompt window.

## At the prompt,

### set the path to the folder containing the JDK compiler:

**set path=C:\Program Files\Java\jdk1.8.0\_45\bin**

### If the prompt is not H:, change it by typing **H:**

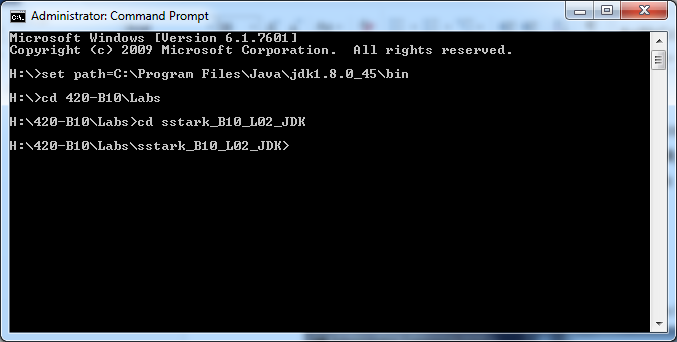
### change the current folder to your labs folder:

**cd 420-B10\Labs**

### change to this week's lab folder:

**cd *username*\_B10\_L02\_JDK**

The results should be similar to the screen shown here:



*To avoid having type the DOS command lines in step 2 every time you start the Command Prompt to compile a Java program, you can create a batch file and execute it. To do this:*

1. *Type the following two lines in Notepad:*

**set path=C:\Program Files\Java\jdk1.8.0\_45\bin**

**cd 420-B10\Labs**

1. *Save the file on your home drive (H:) as* **jdk.bat***.*
2. *Next time you go to the Command Prompt to compile a Java program, type:*

**H:\jdk**

*and hit* **Enter***. This will cause the command lines to execute.*

## Compile the program:

**javac Heritage.java**

***Note***: It will take a few seconds to compile. Be patient.

You should be back at the **>** prompt.

## If there are any error messages, try to figure out what they mean and change your program accordingly in **Notepad**. Then save and recompile your program until there are no error messages.

***Note***: To identify error messages, compare the spelling and case of each word on the highlighted line with the text on the previous page of this lab. Examples of common errors are shown here.

**Examples of Common Errors**

**1. A misspelled word:**

***Message:***

Heritage.java:14: cannot resolve symbol

symbol : class out

location: package Sytems

Sytems.out.println("J8Y 6T3");

^

1 error

***Problem:*** *The word System has been spelt Sytems. Notice that the ^(caret) points to the period following the word that the Java compiler could not recognize.*

***Solution:*** *Change the word Sytems to System in Notepad. Then save and recompile the program.*

**2. A missing semicolon (;):**

***Message:***

Heritage.java:12: ';' expected

System.out.println("325 boul. Cite des Jeunes")

^

1 error

***Problem:*** *A semicolon is missing at the end of the line shown. Notice that the ^(caret) points to the space where the semicolon should be.*

***Solution:*** *Add the semicolon to the end of the line shown in the message in Notepad. Then save and recompile the program.*

# Executing a Java Program

Purpose: Learn how to execute a Java program using the JDK.

To Do:

## Execute your compiled program by typing:

**java Heritage**

at the **>** prompt. Be sure that you capitalize exactly as you did in your source program.

The address of Heritage College should be displayed on the screen.

## To see what files are in your folder, type:

**dir Heritage.\***

How many files are listed? 2 files

What are the file extensions? Heritage.class and Heritage.

**Explanation**:

You created the **Heritage.java** file using **Notepad**. This is the **source code** for the program.

The **Heritage.class** file was created by the **javac** compiler. The **Heritage.class** file contains the **bytecode** for the program. When you execute a Java program you are executing the bytecode.

# The println() and print() Methods

Purpose: Learn how to use the System.out.print() and System.out.println() methods to output data on the screen.

To Do:

## In **Notepad**, open the **Heritage.java** source file if it is not already open. Change the line

**System.out.println("Gatineau, QC");**

to:

**System.out.print("Gatineau, QC ");**

## Save the **Heritage.java** source code.

## In the **Command Prompt** window, *recompile* ***(javac)*** and *execute* ***(java)***the **Heritage** program again.

**Tip**: You can use the up and down arrow keys to recall previous commands so that you don’t have to retype them.

What is the difference between the **println** and **print** methods?

Println will print on a new line

# Identifying and Correcting Syntax Errors

Purpose: Learn how to read the java syntax error messages and correct syntax errors in Java programs.

To Do:

## In Notepad, open the **Heritage.java** source file if it is not already open. Change the line

**System.out.print("Gatineau, QC");**

to:

**system.out.print("Gatineau, QC ");**

## Save the **Heritage.java** source code.

## In the **Command Prompt** window, recompile **(javac)** the **Heritage** program. You should get the following message:

H:\420-B10\Labs\sstark\_B10\_L02\_JDK>javac Heritage.java

Heritage.java:17: package system does not exist

system.out.println("Gatineau, QC");

^

1 error

*The first line contains the error message: "*package system does not exist*".*

*The next 3 lines give the location of the error. Notice the ^ pointing to the location of the error. The message states that the symbol cannot be resolved and the ^ indicates that the symbol that it can't resolve is the one immediately preceding the ^ - in this case the word* ***system****. Because Java is case-sensitive, it does not perceive system and System to be the same word and while it knows what* ***System*** *is, it does not know what* ***system*** *is.*

## Switch back to Notepad and change the line

**system.out.print("Gatineau, QC");**

to:

**System.out.print("Gatineau, QC ")**

## Save the change and recompile the program in the Command Prompt window. What is the error message this time?

There’s a missing colon at the end of the line.

Where is the ^ pointing at? Where the colon should be at the end of the line.

***On Your Own:setc:***

## Create a new Java source file called ***username****\_***MyAddress.java** containing a class called ***username****\_***MyAddress** that displays your name and address on 3 lines. Save it in your **420-B10\Labs\*username*\_B10\_L02\_JDK** folder

## Compile and execute your program.

# Java Variables and the Assignment Statement

Purpose: Learn to declare and initialize variables in a Java program.

To Do:

## Create a new file in **Notepad** and type in the following program:

public class SimpleAdder

{

public static void main (String [] args)

{

int num1, num2, sum; // variable declaration

num1 = 14; // variable assignment

num2 = 7;

sum = num1 + num2;

System.out.println ("The sum of "+ num1 + " and "

+ num2 + " is " + sum); // display

// results

} // main()

} // SimpleAdder

## Save the file in your **420-B10\Labs\*username*\_B10\_L02\_JDK** folder as **SimpleAdder.java**.

## Compile and run the program in the **Command Prompt** window. What is output? “The sum of 14 and 7 is 21”

## Return to **Notepad** and assign **1.0** to **num1**. Assign **3.0** to **num2**. Save your changes. Compile the program. What happens? Error occurs.

## Return to **Notepad** and change **num1** to **1** and **num2** to **3**.

## Return to **Notepad** and add the following line after the **int num1, num2, sum;** line:

double dnum1, dnum2, dsum;

## Initialize **dnum1** to **1.0** and **dnum2** to **3.0**. Repeat the statements that add and display the **sum**, replacing the **int** variables (*num1, num2, sum*) with the **double** variables (*dnum1, dnum2, dsum*). Save your changes.

## Compile and run it again. Are the two sums the same? Yes

## Return to **Notepad** and make the following changes:

### add another **int** variable called **quotient** and another **double** variable called **dquotient**.

### Assign **num1/num2** to **quotient**.

### Assign **dnum1/dnum2** to **dquotient**.

### Add two **System.out.println()** statements to display **quotient** and **dquotient** in an appropriate sentence.

## Save your changes and compile and run the program again. There should be four lines of output: the sum of the integers, the quotient of the integers, the sum of the doubles and the quotient of the doubles. What is output for **quotient**? **0**

What is output for **dquotient? .3333333333…**

Change **dnum2** to **12345678.99** and **dnum1** to **10**. Save the changes. Compile and run the program again. How is **dnum2** written in the output?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Graphics Supplement (Optional)

Purpose: Learn to create an applet that draws a simple shape.

To Do:

## In Notepad, type the following code:

import javax.swing.JApplet;

import java.awt.Graphics;

public class HappyFace extends JApplet

{

public void paint(Graphics canvas)

{

canvas.drawOval(100,50,200,200);

canvas.fillOval(155,100,10,20);

canvas.fillOval(230,100,10,20);

canvas.drawArc(150,160,100,50,180,180);

} // paint()

} // HappyFace

## Save the file as **HappyFace.java**.

## In the **Command Prompt** window, compile **HappyFace.java**. Try to run it. What happens? Tells me I have to redefine the main method.

**Explanation**:

An applet is a GUI program that runs in a web browser. Therefore you cannot run it in command prompt mode.

* 1. Create a new file in **Notepad** and type in the following:

<applet code="HappyFace.class" width=350 height=300>

<\applet>

## Save the file in your **420-B10\Labs\*username*\_B10\_L02\_JDK** folder as **HappyFace.html**.

* 1. To see what the program actually does:
     1. Start the **Opera** browser.
     2. Select **Open** from the **File** menu.
     3. Click **Browse** from the Open dialog box.
     4. Select your **H:** drive in the drop down box. Double-click on the **420-B10** folder, then again on your **Labs** folder and finally on your ***username*\_B10\_L02\_JDK** folder.
     5. Select **HappyFace.html** and click **Open**. You may have to specify that you wish to allow blocked content. If you get a security error, close the browser and do the following:

#### Open the **Control Panel**

#### Search for **Java**

#### Open it and click the **Security** tab.

#### Select the **Edit Site List** button.

#### Click **Add**

#### Add the following file to the site list <file:///H:/420-B10/Labs/username_B10_L02_JDK/HappyFace.html>.

#### Click **Add**.

* + 1. Click **OK** to close the **Open** dialog box. What is displayed? A happy face

## Try to modify your happy face to put on eyebrows, ears or a nose.

# To finish:

1. Zip your **H:\420-B10\ Labs\*username*\_B10\_L02\_JDK** folder. To zip your folder:

### Start **Windows Explorer** by clicking on **My Documents** on your desktop.

### Right click on your **H:\420-B10\ Labs\*username*\_B10\_L02\_JDK** folder name.

### Select **Send to 🡪 Compressed (zipped) folder**.Hit **Enter**.

### Start **Internet Explorer** and log in to **Moodle**. Go to the **420-B10** course page. Click on **Lab 2 submission** and follow the instructions to upload your zipped folder.

## Do the **Lab 2 Review Quiz** in **Moodle**.