

What Is Java?

Java is a computer programming language that enables programmers to write computer instructions using English based commands, instead of having to write in numeric codes. Java is known as a “high-level” language because it can be read and written easily by humans. Like English, Java has a set of rules that determine how the instructions are written. These rules are known as its “syntax”. Once a program has been written, the high-level instructions are translated into numeric codes by compiling the source code so computers can understand and execute the code.

Moving to Java

Before we can begin to write code you must install the JDK (Java Development kit). The JDK contains the JVM (Java Virtual Machine) that enables you to run Java programs on your computer. The JDK contains tools such as the compiler javac. Javac converts the “syntax” into the numeric code, the “.class” file so the computer can understand and execute the code. This process will be discussed later. The javac command can be executed from the command line, but for ease of use, we will be using the javac command in TextPad. If you install the JDK first and then Textpad this command will show up automatically, otherwise you will have to add the command manually to TextPad.

- Read the first 3 pages <http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>
- Install the JDK (Java Development Kit) Can be downloaded from Sun.com <http://java.sun.com/javase/downloads/index.jsp> or install from <Http://plourde.smccme.edu> CMPT 125
- Install TextPad <http://www.textpad.com/download/> or install from <Http://plourde.smccme.edu> CMPT 125.

Note: Install in this order: 1. JDK, TextPad

PATHPATH and CLASS

The following is a general programming errors, which most beginning java programmer will come across. Here is a solution on how to solve the problems when running on a Microsoft Windows Machine.

1. ‘javac’ is not recognized as an internal or external command, operable program or batch file

When you get this error, you should conclude that your operating system cannot find the compiler (javac). To solve this error you need to set the PATH variable.

How to set the PATH Variable?

Firstly the PATH variable is set so that we can compile and execute programs from any directory without having to type the full path of the command. To set the PATH of JDK on your system Windows XP and Vista, add the full path of the `jdk<version>\bin` directory to the PATH variable. Set the PATH as follows on a Windows machine:

- a. Click Start > Right Click "My Computer" and click on "Properties"
- b. Click Advanced > Environment Variables.
- c. Add the location of bin folder of JDK installation for PATH in User Variables and System Variables. A typical value for PATH is:

C:\jdk<version>\bin (jdk<version is nothing but the name of the directory where jdk is installed)

If there are already some entries in the PATH variable then you must add a semicolon and then add the above value (Version being replaced with the version of JDK). The new path takes effect in each new command prompt window you open after setting the PATH variable.

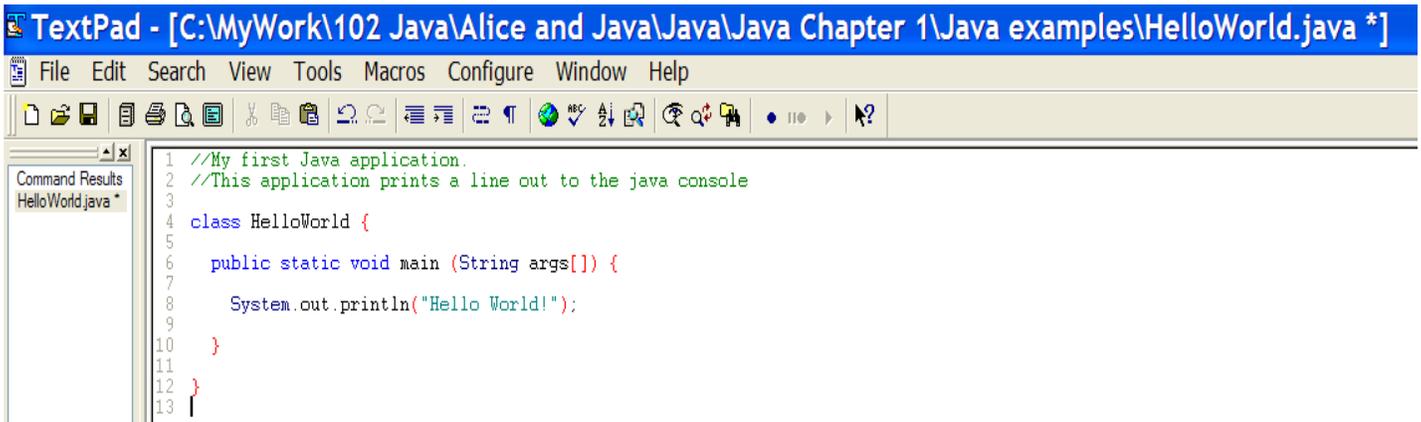
Create Your First Java Application

- Open TextPad and type the following as it appears below:

```
//Your name  
//My first Java application. 10/13/2010  
//This application prints a line out to the java console
```

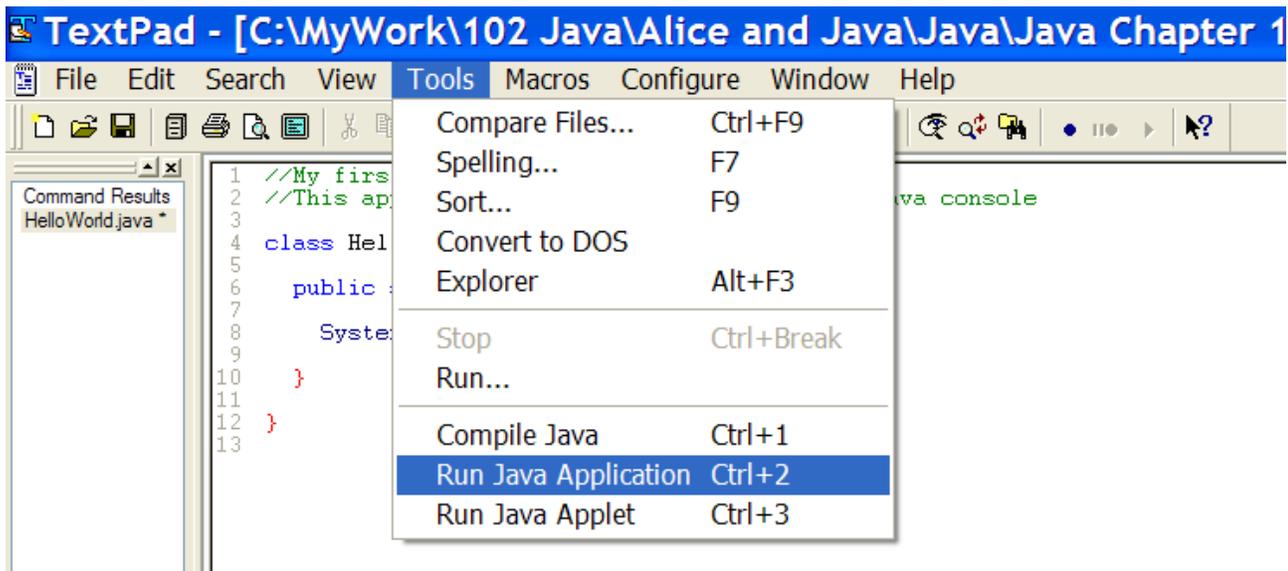
```
class HelloWorld {  
    public static void main (String args[]) {  
        System.out.println("Hello World!");  
    }  
}
```

- Note the colors. You will not see the colors until you save your file as a .java
- Save your file now as **HelloWorld.java**. This must be the same name as the class name as on the 4th line that you had typed. Case does matter! Helloworld, helloworld and HelloWorld are all different.
- Note the green text that starts with // are comments. Please always comment all of your programs with your name and project. Your code in TextPad should look like the image below.

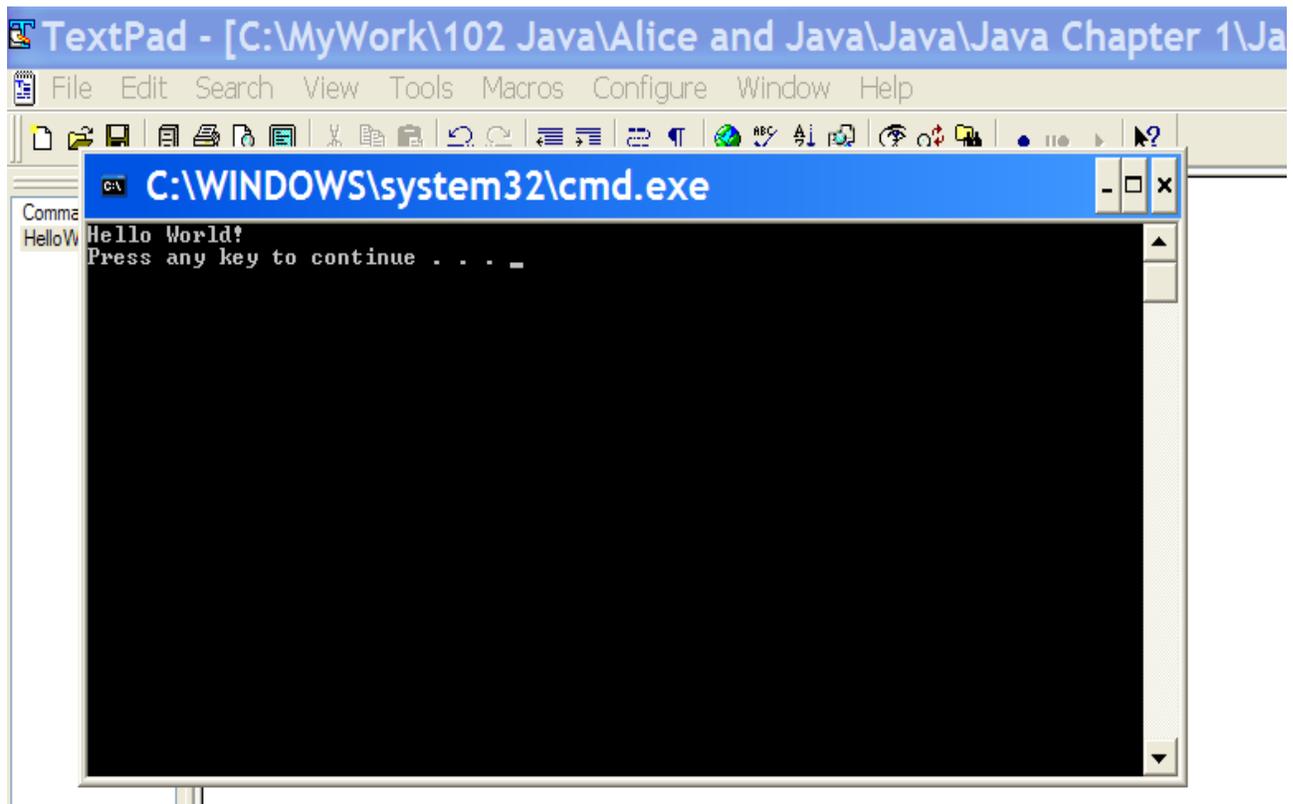


Compiling and Running HelloWorld.java

- Compile **HelloWorld.java** By going to Tools...Compile Java. There should be no errors. If you have errors review the code above to see if you have forgotten anything.
- Run the **HelloWorld.java** by going to Tools...Run Application as shown in the image below.



- The text will appear in the java console as shown in the image below.



So what does it mean?

About the HelloWorld Program

I created a class named “HelloWorld” containing a simple main function within it. The keyword **class** specifies that we are defining a class. The name of a public class is spelled exactly as the name of the file (Case Sensitive). All java programs begin execution with the method named main(). Main method that gets executed has the following signature: **public static void main(String args[])**. Declaring this method as **public** means that it is accessible from outside the class so that the JVM (Java Virtual Machine) can find it when it looks for the program to start. It is necessary that the method is declared with return type **void** (i.e. no arguments are returned from the method – does not return anything). The main method contains a String argument array that can contain the command line arguments. The brackets { and } mark the beginning and ending of the class. The program contains a line ‘**System.out.println(“Hello World”);**’ that tells the computer to print out one line of text, ‘**Hello World**’. The semi-colon ‘;’ ends the line of code and is also referred to as a terminator. The double slashes ‘//’ are used for comments. Comments should be used at the beginning of your program with your name, date and intent of the code. Comments can also be used to describe what the source code throughout the program. Everything to the right of the slashes on the same line does not get compiled, as they are simply the comments in a program.

Java Main method Declarations

```
class MainExample1 {public static void main(String[] args) {}}
class MainExample2 {public static void main(String []args) {}}
class MainExample3 {public static void main(String args[]) {}}
```

All three are examples above are valid main methods that accept a single String array argument.

A Java Application

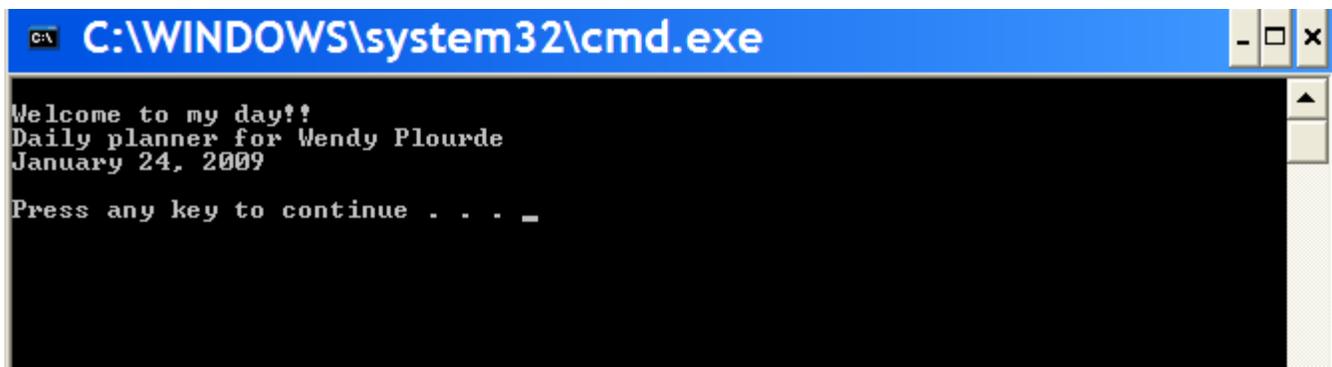
Look at the code below. Create the application with your name by retyping the code below.

- Note the comments in this example are different than the HelloWorld application. The Welcome application is demonstrating how to comment a block of text rather than doing it line by line by using `/* */`.
- Look at each line of code in the main method. Each line has to be terminated by the `;`.
- Remember to save your file with the same name as the class name on line 11: in this case **Welcome.java**. Note: Case does matter.
- Pay attention to the color scheme in TextPad. Parentheses and brackets are always red. Comments and text/string values are green. Key words in the class and methods are blue. The colors are clues to mistakes you may have made typing in your code. When a word is black and is supposed to be blue, check your spelling. Note: In the `.println` command the character after the `"t"` in `print` is an `"l"`, as in **lion** and not the numeric value `"1"`.

```
1  /*
2  Chapter 2: Welcome to my day
3  Programmer: Wendy Plourde
4  Date January 5, 2008
5  File Name: Welcome.java
6  Purpose: This is a project that will display a welcome message,
7  the user name and the system date.
8  */
9
10
11 public class Welcome
12 {
13     public static void main(String[] args)
14     {
15         System.out.println();
16         System.out.println("Welcome to my day!!");
17         System.out.println("Daily planner for Wendy Plourde");
18         System.out.println("January 24, 2009");
19         System.out.println();
20     }
21 }
22
```

Compile and Run the Welcome Application

When running the application the results will look like the figure below.



The screenshot shows a Windows command prompt window with the title bar `C:\WINDOWS\system32\cmd.exe`. The window contains the following text:

```
Welcome to my day!!
Daily planner for Wendy Plourde
January 24, 2009

Press any key to continue . . . _
```

Another Java Application Using java.util

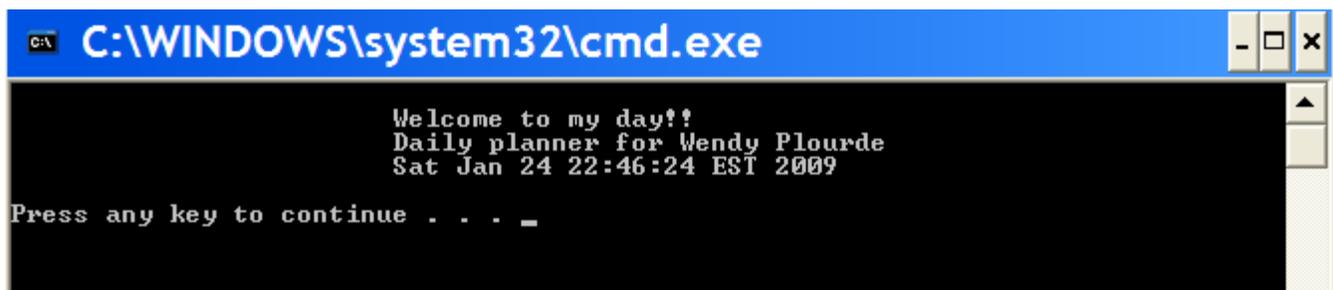
Java.util Provides classes for the date, time, timer and calendar functionality. Change the Welcome application to import java.util and print out the system date. Your code should look like the code below.

- Note the differences in the WelcomeDate application from the Welcome application above.
- The import line has to go before the class
- The System.out.println uses \t for tabs to center the text.
- The use of the date constructor to construct the date.
- The use of the “+” is to concatenate (join together) the text in the double quotes and the key word currentDate

```
1  /*
2  Chapter 2: Welcome to my day
3  Programmer: Wendy Plourde
4  Date January 24, 2009|
5  File Name: Welcome.java
6  Purpose: This is a project that will display
7  a welcome message, the user name and the
8  system date.
9  */
10
11 import java.util.Date;
12
13 public class WelcomeDate
14 {
15     {
16         public static void main(String[] args)
17         {
18             Date currentDate = new Date(); //Date constructor
19             System.out.println();
20             System.out.println("\t\t\tWelcome to my day!!");
21             System.out.println("\t\t\tDaily planner for Wendy Plourde");
22             System.out.println("\t\t\t" + currentDate);
23             System.out.println();
24         }
25     }
```

Compile and Run Application

After compiling and running the application the results will look like the figure below.



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The output of the Java application is displayed as follows:

```
Welcome to my day!!
Daily planner for Wendy Plourde
Sat Jan 24 22:46:24 EST 2009

Press any key to continue . . . _
```

Creating an Applet

An applet is used to run in the browser or on the internet. Creating an applet involves importing java.awt (Abstract Window Toolkit (AWT)) provides classes for creating user interfaces and for painting graphics and images. You also have to create an HTML file to call your class.

- Review the code below
- Change the WelcomeDate to the WelcomeApplet below
- You will need to import the package java.applet so the applet can communicate with the content.
- This applet use a graphic called "smile.gif" you may use any graphic you want. You may need to make an image smaller or choose small image to fit your applet
- This applet uses a drawString instead of a println as the Welcome application did.
- This applet uses the paint method and assigns it to the letter "g".
- The applet uses x and y coordinates for the positioning of text and graphics
- The applet uses a setBackground() method
- After coding the WelcomeApplet compile it the same way you compiled the Welcome application. Fix any errors and compile again
- Then create the HTML file below

```
1  /*
2  Chapter 2:  Welcome to My Day
3  Programmer: Wendy Plourde
4  Date:      January 25, 2009
5  Filename:  WelcomeApplet.java
6  Purpose:   This project displays a welcome message, the user's
7             name, the system date, and an image in an applet.
8  */
9
10 import java.util.Date;
11 import java.awt.*;
12 import java.applet.*;
13
14 public class WelcomeApplet extends Applet
15 {
16     public void paint(Graphics g)
17     {
18         Date currentDate = new Date(); // Date constructor
19         g.drawString("Welcome to my day!",200,70);
20         g.drawString("Daily planner for Wendy Plourde",200,100);
21         g.drawString(currentDate.toString(),200,130);
22         Image smile; // declare an Image object
23         smile = getImage(getDocumentBase(), "Smile.gif");
24         g.drawImage(smile,15,15,this);
25         setBackground(Color.cyan);
26     }
27 }
```

HTML File

- Create a new document in TextPad and type the following code
- On the Applet Code line make sure your file matches what you called you class
- The .class file is created when you compiled WelcomeApplet.java

- Save this file as an WelcomeApplet.HTML file
- Note the color coded tag do not appear until the file is saved.

```
1 <html>
2 <applet code = "WelcomeApplet.class" Width = "400" height = "200">
3 </applet>
4 </html>
```

Running the Applet

When running an applet you are running the HTML file which is calling the .class file or compiled java code.

- To run the applet select Tools...Run Java Applet
- The HTML file should be selected if it is in the same directory/folder where your .java and .class files are.
- Your result should look similar to the example below



Basic Definition

Syntax – Java's set of rules that determine how the instructions are written

JDK – Java Developers Kit Contains commands, tools and documentation for Java programmers

JVM – Java Virtual Machine interprets Java language for computers

Javac – The compiler command puts syntax into machine language

Classes and Methods

class - In the real world, you'll often find many individual objects all of the same kind. There may be thousands of other bicycles in existence, all of the same make and model. Each bicycle was built from the same set of blueprints and therefore contains the same components. In object-oriented terms, we say that your bicycle is an *instance* of the *class of objects* known as bicycles. A *class* is the blueprint from which individual objects are created.

method - is a set of Java statements which can be included inside a Java class

Date - Class that contains date and time

main – “main” is a method

system.out.println – Method on how the program is handling out put

\t – Tab spacing used for formatting

+ - Concatenation used to join values and text together

drawString - Method on how the program is handling out put

setBackground – Method used to tell the program how to set the background

Keywords - Java reserves some words. They cannot be used as variable, method or class names.

Java Keywords				
abstract	do	import	public	throws
boolean	double	instanceof	return	transient
break	else	int	short	try
byte	extends	interface	static	void
case	final	long	strictfp	volatile
catch	finally	native	super	while
char	float	new	switch	
class	for	package	synchronized	
continue	if	private	this	
default	implements	protected	throw	
Reserved keywords (not currently in use)				
const	goto			
Reserved Literals				
null	true	false		

public – Keyword defining the scope of the class

void – Keyword meaning nothing is returned

import – Keyword to use a class in a package

Packages

Java.util – A collection of classes that contain utilities such as date and time.

java.awt - (Abstract Window Toolkit (AWT)) provides classes for creating user interfaces and for painting graphics and images

java.applet – Enables the applet can communicate with the content

Assignments

1. Read the first 3 pages <http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>
2. Install the JDK (Java Development Kit) on your home computer.
3. Install Textpad

Programming Assignments

1. Create the HelloWorld application.
2. Create the Welcome application
3. Create the WelcomeDate application
4. Create the WelcomeApplet