

CIS163AA - Java I

Glendale Community College

Instructor: Gary Marrer

Lesson One



This Week's Objectives:

Topics:

Class Into: Introduction to Class (syllabus, class calendar, rules, format)

History of Java

Topics covered in class

Getting Class Software

First Java application

Using Textpad to create and compile Java

Using Java Web Site

Welcome to Class!

Over the next sixteen weeks, we will become partners in learning Java. The class will use a new learning format called hybrid. Hybrid means the class will be a combination of both on-line and in-class instruction. The bulk of the learning process will take place on-line through a course management tool called BlackBoard (a.k.a. Bb). We will also have five class meetings where you will get the opportunity to address questions to me directly and interact with other students in the class...

I will provide the foundation (via lectures, assignments and tests) and each of you will read the lectures, textbook, and complete the exercises, assignments and tests. The content delivered to you on the Internet is the same as the content delivered in my traditional classroom course. However, instead of me being able to call on you with questions, we will be learning asynchronously where the timing of our interaction will no longer be real-time. Our communication and discussion may take place the same day

or maybe the next day. We will need to keep this in mind as we send information to Bb and communicate with others.

The sequence of topics covered in class is roughly the same sequence in the textbook. For that matter, the textbook will drive a lot of what is discussed in class. I will not repeat the book in my lecture but may I elaborate or expand on key topics. I will refer to the book in my examples and identify key sections in the book that cover class topics. The textbook is an integral part of hybrid learning. In addition, I have a couple of lessons which I will present my own material. These lessons will deal in more detail the concepts important to Object Orientated Programming as it pertains to Java.

On-Site Class Meetings

We will have three class meetings. Dates and times for our GCC class meetings can be found in the course schedule and on my faculty web page calendar (<http://web.gccaz.edu/~gmarrer>)

Note: 1st class Meeting Monday January 26th 7:00 pm to 9:00 pm HT2-156

Class Software: Please note that this class will be using Textpad for our application development. Textpad is a shareware programming editor and is a free Java IDE (it is included on your Textbook CD-ROM). We will also introduce other IDE's such as BlueJ, NetBeans, JCreator and Eclipse. The one thing about Java development is that there are a lot of choices and you will find that each IDE has its own strength and weakness. No one IDE is perfect and it is up to you to decide which one you would like. For the beginning of class, we will stick to TextPad since it will be easier if we all work with the same development product.

Even though we are on line, I thought you might want to know what I Look Like:



On-line classes (past experience)


I have taught on-line classes for several years using several on-line educational tools (BlackBoard, WebCT, WebBoard, homegrown (with HTML/JavaScript/Java applets), Midas, etc.). I have also been an on-line student myself. These on-line learning tools are the same and are different from traditional classes. They are the same in that both on-line and traditional class formats use a textbook, tests, lectures along with tests and labs to teach you Java. They are different in that on-line classes require the motivation to work at readings and assignments daily. [Many believe that on-line classes are easier and require less effort. This is a fallacy. There is more flexibility with on-line classes but not less work.](#) One of the primary success factors for any programming class is to [stay up with the reading assignments.](#) This is critical. You will have weekly quizzes to test your comprehension of the chapter and I also ask that everyone sends me a weekly status email.

Before we start!

One requirement to participate on-line classes is that you need to know how to use the tools. During our first on campus class meeting at GCC, we will review much of this information together as a group. Please look over Bb documentation and review the topics in this lesson and become comfortable with the class tools we will be using. Do this the very first week. If you can not attend one of our on-site GCC evening classes (1st Meeting is Monday January 26th 7:00pm to 9:00 pm HT2-156) you will have access to an .MP3 audio of the class and you will be required to write a one page paper summarizing topics covered.

Questions.... Questions.... Questions

I love questions. If you have a question that you don't mind sharing with the class enter that question into the one of the question topics in our Bb Discussion Tool. If you have a question, which you would like to ask me and not share with the class, please send it along as email. I will return responses to email sent to my personal account within 24

hours. I have placed on my lecture web pages several Ask Me icons  which when clicked will open up email with my address so that you can send me your question. Try it and send me an email now!

Bb Down!

Not to worry, most of the materials are available via links from my faculty web page (just click on the CIS163AA Link) The grade book and discussion areas will not be available but just about everything else has an alternate path.

Format:

Each week the lecture document will provide the roadmap for the rest of the week. The topics found in this lecture document will remain more or less the same.

Objectives for week (Each week has a different set of objectives)

Lecture (lecture weekly) - main points reviewed

Assignments (all assignments are accessible on the calendar on the Class web page)

Book highlights

My Turn: experiences in the field to support the class objectives

Assignments (not every week but for most)

Documents from Bb will be formatted in a combination of .PDF and .HTML files. I try to create .PDF files for those documents a student is likely to print (i.e. syllabus, class schedule and lessons).

Grading

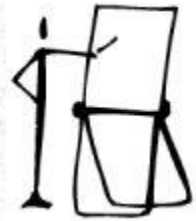
There is nothing very scientific about the grading for this class. There are a total of 560 points (see Syllabus for the breakdown of assignments and tests) for which you receive more than 90% you will receive an A, 80% a B, 70% a C and so on. You receive points for taking tests, participating in class (quizzes and discussion questions) and turning in lab assignments. **To receive an A for the class you must have taken ALL tests and turned in all labs. All labs must have a score of 80% or better. No Exceptions.** Grades will be posted in the Bb grade book.



CIS163AA - Java I

The goal of this class is to give you an introductory understanding of the Java programming language. The topics covered come from a set of core competencies developed by the Maricopa Community College District to insure that all CIS163AA students finish the course having covered the base topics (the competencies document can be accessed via the on-line class schedule on a link in the top right section of the course listing or at link (<http://www.maricopa.edu/curriculum/A-C/072cis163aa.html>)). Java is a complicated but robust language that can create standalone applications,

applications launched from a browser and applications that can have a Windows like interface.



Lecture

Chapter 1

History of Java

From a historical standpoint, the Java programming language has a rich past. Born out of some of the original enthusiasm of the dot net boom, Java was originally designed as a programming language by Sun Microsystems for a new hardware device (called “set top”) for connecting our homes to the Internet. These devices would be installed with your television and use your cable connection for Internet connectivity. Needless to say, most of this may get our Internet to the cable company but we do not access the net via the television or this proposed Internet “set top” device (at least not yet). Since the “set top” technology did not take off, Sun Microsystems was left with a programming language that had no home.



The original Sun Set Top TV

Fortunately, at about the same time the Sun was working on Java, Netscape was in need of a programming language which could enhance the Netscape browser with a programming function. What was originally called Oak by Sun is now called Java and

James Gosling and his team of programmers found an initial home for their Java programming language. There has been much written about the development of Java.



James Gosling (“Father of Java”)

Having celebrated a five-year anniversary a couple years back, the people at Sun created a web page which summarized the development of Java. It is interesting from an educational perspective since it describes how new technology is taken from planning to implementation. If you are interested in creating your own programming language or developing that new technology to become the next Bill Gates, this might make interesting reading.

History of Java by Sun <http://java.sun.com/features/1998/05/birthday.html>

Java Programs (Three Flavors)

In this class, we will actually develop three different kinds of Java programs. We will start by creating what's called a **console application**. This console application is executed from the Windows (or Linux or Apple, or Palm or etc.) command line or what is more commonly called the DOS prompt. This application is one which will look very crude and unsophisticated as compared to what we're used to seeing in a Windows, Macintosh or Linux environment. The next type of application which will develop is one that was more familiar with. This **Java Frame** application uses a **GUI interface** that has controls command buttons, combo boxes, text boxes, etc.,) we are accustomed seeing in products like MS-Word and MS-Excel. The last Java applications we will create use the GUI controls of Java coupled with the Internet and a browser. A Java **Applet** is a Java program written to run within the browser using the controls of the browser along with controls of the applet to perform application tasks. Java applications exist in all three forms. You are probably most familiar (or maybe didn't realize it) with Java applets since these are executed automatically from many Web pages. It is important understand that Java is also used just as Visual Basic.Net, C#.Net C++ and other traditional programming languages for the development of business applications. Because Java contains all the same functionality and flexibility as the more traditional Windows development languages, it is almost impossible to tell when you are running an application written in Java vs. one written in VB, C# or C++.



Write Once Run Anywhere

With the Java, you have support for a write once run anywhere environment. What this means is that your compiled Java source code (a class file) can be moved to any environment that supports a Java Virtual Machine (JVM). There are some qualifications however; first of all, there are different versions of Java and if you write a program using the newer release of Java (currently 1.6) then you may be required to update the Java Virtual Machine ([free from Sun](#)) to run this new application. In addition, one of the differences between operating systems which are difficult to solve with any programming language, is differences associated with displayed screen output. I think everyone would agree that the display options and the objects that can be displayed is much greater on a Windows operating system then the display on a wristwatch or cell phone (both of which can support a Java Virtual Machine). If you develop for other environments besides Microsoft Windows, you'll find that there may be some restrictions you must follow when developing for your environment. In the case of a PDA operating system like Palm OS, you would need to use classes specific to the operating system to support the graffiti input and touch sensitive screen of the palm computer.

Steps in Development of a Java Application

Application development and Java is very similar to other programming languages. The process typically followed is:

1. For starters, the programmer must start the application development process off with a flowchart and pseudo code of the logic needed in the application. Like every good builder, you do not want to start building a house before you have a blueprint to go by.
2. Next we will use a text editor (in our case Textpad) to type in the Java programming statements that represent the logic steps of the program.
3. Once the steps are added to a text file and saved with a .java file extension, the program can be evaluated by the compiler. The compiler looks each line of the program and determines if the syntax of the programming statement is valid and if objects and variables identified in the program are valid. If it finds any errors it reports back to the user the line number which is an error along with a short description of that error.
4. When a program compiles successfully it creates a new file with the same name as the original source file but with a file extension of .class. This class file is comprised of byte codes. Byte codes are Unicode characters which when read by a Java Virtual Machine and will execute the instructions code in your source file. The class file that created in the compile process can be used on any operating system which has a Java virtual machine.

5. From TextPad (other IDE's also support this), you can select Tools and run your Java Application or Applet to test it and make sure it is performing the required tasks. If not, you will return to the source code and start the process over again.

As of this date, Java virtual machines have been created for everything from a smart card to a personal computer to a personal computer running Windows, McIntosh or Linux operating systems. It is important remember the Java Virtual Machine is not mechanical. It is simply a software program which converts the byte codes .class file to machine codes understandable by the operating system is being run on. In short, the Java virtual machine is nothing more than an interpreter or translator converting by byte codes into language the operating system to understand.



Fixing Compiler Errors

Sometimes the short compiler errors are not as helpful as it could be. Learning how to evaluate and repair errors found by the compiler is a task that takes a lot of practice and with experience becomes almost second nature. For those were new to Java, fixing syntax errors to be very frustrating activity. Once the program has compiled without any errors it is ready for execution. Important Tips to successfully compiling in executing your first Java program

Common Problems in Creating your Program Code

- When you open up Textpad and you typed the java *class* statement, make sure the name of the class matches in the name of the file containing the source code. For example, if your Class statement identifies class as *Assignment1*, make sure that you save the file in TextPad as *Assignment1.java*,
- If you do not save the file with the same name as the class, the program will not compile. If you forget to use a file extension of .java, the programming statements in your application will not change color to reflect the type of statement is. The color of the statements based on function is call syntax Color highlighting and is available with most IDE's. Syntax coloring is a big help in spotting potential errors before compilation.
- Remember, Java is case sensitive. What this means is that a reference to a variable called total is different than another variable called a TOTAL because the case of the letters are different.



Our First Java Program

I have created a video lesson to show how to copy and paste this code into TextPad and



then compile and test it. You can see this by clicking on this Video Lesson.

<http://web.gccaz.edu/~gmarrer/cis163aaBb/videolessons/VLFirstApp/VLFirstApp.html>

```
// First Java Application
// Gary Marrer
// CIS163AA
```

```
import javax.swing.*; // import the swing package to use JOptionPane
object
```

```
public class FirstJavaApp
{
```

```
    public static void main(String args[])
    {
```

```
        String zoneCost, weightCost, sizeCost;
        int zone, weight, size, total;
```

```
        zoneCost = JOptionPane.showInputDialog("Enter Zone Cost");
        weightCost = JOptionPane.showInputDialog("Enter Weight Cost");
        sizeCost = JOptionPane.showInputDialog("Enter Size Cost");
```

```
        zone = Integer.parseInt(zoneCost);
        weight = Integer.parseInt(weightCost);
        size = Integer.parseInt(sizeCost);
```

```
        total = zone + weight + size;
```

```
        JOptionPane.showMessageDialog(null, " total cost " + total ,
        "Answer" , JOptionPane.INFORMATION_MESSAGE);
```

```
        System.exit(0);
```

```
    } // end of main method
```

```
} // end of FirstJavaApp class
```



Getting Help...

Depending upon the IDE use to develop the job applications, you may have one or many options for getting Java help. When using the TextPad environment, you'll have to use the Java API Page (<http://java.sun.com/javase/6/docs/api/>) accessed over the Internet. It would be a good idea to bookmark this page as you will probably be using it frequently class.

Eclipse and other IDE's like BlueJ and Sun NetBeans include built-in help (Local access to the Java API documentation) along with Intellisense support which will help you fill in syntax automatically.

Experience with other hybrid classes has shown that video lessons (video capture of me walking students through a problem with my PC accompanied with my voice) have been very well received. I have several topics already identified and will introduce other video lessons as needed.

Note: Typos, Broken Sentences, etc..

This is the first time CIS163AA (Java) has been taught as a hybrid class in Bb. I have done my best to locate all of the typo's and broken sentences but there will always be a couple that sneak through. If you find one, let me know via email and I will do my best to get it fixed as soon as possible.



Book Notes:

This is the first time we are using the Shelly - Cashman textbook. Course Technology and the Shelly - Cashman series have a long reputation for excellent coverage of

computer topics. This book seemed to be especially appropriate for the hybrid class due to the amount of clipart and photos. This can be very important when the student is left on their own for a large portion of the information covered. I will also supplement the reading and lectures with video lessons that will bridge the gap on more complicated topics. We will start officially next lesson but I would start going through the textbook as soon as possible.



MY Turn:

IDE's Integrated Development Environment

With a Java we have several IDE options. Remember IDE stands for **Integrated Development Environment**. Essentially this is software that allows us to create source code (or Java statements), compile the code into a class file, and then finally taking that class file of byte codes and execute it in a Java virtual machine. The fact that we can develop our code, check our code for syntax errors and test our code in one application makes the IDE a very important tool in programming. For the first assignments, we will be using Textpad. Textpad and is essentially a text editor like Microsoft Notepad but with the additional capability of allowing us to compile and execute our Java programs. As you develop Java applications, you'll return several times to the source editor to fix errors (syntax, logic and runtime).

As we get deeper in the course in work more extensively with objects, we will introduce another IDE, specifically BlueJ. BlueJ is a free open source program that was developed by University of Kent (UK) and Deakin University (AU) for teaching Java. It has long been used by Java teachers all over the world <http://www.bluej.org/about/users.html>.





Chapter Review

- Log on to Bb. If you are a new user, set up your account
- Read the syllabus
- Read the course calendar
- Install Java SDK (from the textbook or Sun web site) ([free from Sun](#))
- Install TextPad (from textbook or Textpad.com web site)

Questions?

