



JAVA

THE 1-STEP AT A TIME **JAVA PROGRAMMING**
FOR BEGINNERS GUIDE TO LEARN JAVA

MATTHEW ADAMS

Java

The 1-Step At A Time Java

*Programming For Beginners Guide to
Learning Java*

Copyright 2017 by Matthew Adams - All rights reserved.

The follow eBook is reproduced below with the goal of providing information that is as accurate and reliable as possible. Regardless, purchasing this eBook can be seen as consent to the fact that both the publisher and the author of this book are in no way experts on the topics discussed within and that any recommendations or suggestions that are made herein are for entertainment purposes only. Professionals should be consulted as needed prior to undertaking any of the action endorsed herein.

This declaration is deemed fair and valid by both the American Bar Association and the Committee of Publishers Association and is legally binding throughout the United States.

Furthermore, the transmission, duplication or reproduction of any of the following work including specific information will be considered an illegal act irrespective of if it is done electronically or in print. This extends to creating a secondary or tertiary copy of the work or a recorded copy and is only allowed with express written consent from the Publisher. All additional right reserved.

The information in the following pages is broadly considered to be a truthful and accurate account of facts and as such any inattention, use or misuse of the information in question by the reader will render any resulting actions solely under their purview. There are no scenarios in which the publisher or the original author of this work can be in any fashion deemed liable for any hardship or damages that may befall them after undertaking information described herein.

Additionally, the information in the following pages is intended only for informational purposes and should thus be thought of as universal. As befitting its nature, it is presented without assurance regarding its prolonged validity or interim quality. Trademarks that are mentioned are done without written consent and can in no way be considered an endorsement from the trademark holder.

Table of Contents

[Introduction:](#)

[Chapter 1: The Basics of Java](#)

[Chapter 2: Working with Variables in the Java Language](#)

[Chapter 3: Working with Conditional Statements](#)

[Chapter 4: Iteration Statements and Looping Statements in Java](#)

[Chapter 5: What are Arrays and How Do They Work in Java?](#)

[Chapter 6: Functions and Methods Inside Java](#)

[Chapter 7: Working with Classes and Objects](#)

[Conclusion](#)

Introduction

I want to thank you and congratulate you for purchasing the book, *Java: The 1-Step At A Time Java Programming For Beginners Guide to Learning Java*.

This book contains proven steps and strategies on how to learn how to use the Java programming language. This is a great programming language that you can learn how to use when you want to get into coding, and when you would like to create some of your own webpages for personal and business uses.

While some people are scared that coding will be too hard for them, Java is an easy language to learn and you will be able to learn all the basics with the help of this guidebook. Not only will we take a look at all the basic commands and codes that you need to make Java work, we will combine exercises to the information so you can give these a try.

In this guidebook, we will take a comprehensive look at what Java has to offer. We will talk about how to get started with Java, what the conditional statements are all about and how to use them, how to work with loops and arrays to make more power in your system and to save room when writing the code out, and even how you would work with the variables. All of these are important to know when writing out your first codes in Java and once you master them, the possibilities for writing your own code are endless.

When you are ready to start learning how to use the powerful Java coding language for your needs, make sure to take a look through this book. It will provide all the information you need to learn some of the very basic, and yet very powerful, parts of the Java program so that you are able to write out some great codes.

Thanks again for purchasing this book, I hope you enjoy it!

Chapter 1: The Basics of Java

Sometimes, getting started in the coding language may seem confusing. You may worry that you won't understand how the program works and that you will never be able to figure it out. For the times when you want to learn how to make a web page, you will find that working in Java would be the best option to choose. It is easy, by using some of the latest in coding technology and being based as an object oriented programming language it helps anyone learn how to use it, whether they have some experience in this field or they are starting out brand new.

When working with the Java coding language, you will find that it uses a lot of the features that you will find with C++ as well as C#, but things are going to work slightly differently because you are making a web page out of the system. Java is also not quite as complex as you will find with these other languages, so it will work on many different platforms, is easy for other people to use, and won't be such a hassle when you are first getting started.

The Java programming language is a great option to use when you want to work on online projects. It is set up to be easy to get down, even as a beginner and after looking through some of the tutorials in this guidebook, as well as the codes later on in this chapter, you will be able to create some of your own work in no time.

There is a lot that you will get to enjoy with this kind of language. First, it is free to download and so is the text editors and the environment that you need, so you can give it a try and if you don't like how it works for some reason, it is free and you didn't lose anything but a little time. It is set up for beginners to use as well so you will like how easy it is to get it down.

The software has a large library to help you to create some of your own web pages online so make sure that when you download this program that you go and see how some of the codes will work for your needs. Add in that there is a large community available to ask questions to, that you have many options for creating your own web page, and that this program works on any browser that you want without requiring you to make changes to the code, and you will find that Java is one of the best programming languages to

learn when you want to work on online projects.

Let's take a look at how this language works and what all you will be able to do when learning the Java coding language.

What is Java and Why is it Useful?

Before we get too far into learning some of the codes, it is a good idea to understand what the Java language is all about. Java is a programming language that was originally made to help create a variety of applications right on your computer. It is pretty common for users to go online and then need to use the Java runtime in order to use these programs as they want. It is likely that you already have a form of Java on your computer since it is so widely used. Some people will add the Java plug-in to their computer so that they can run all their apps right on the browser.

One thing to note is that Java is not the same thing as JavaScript. JavaScript is more for working with HTML documents inside of a web page and it isn't good for working on its own because it needs to be connected to a web page to work. This is sometimes confusing to those just starting out, and it is easy to get the names Java and JavaScript mixed up.

JavaScript can be useful for many online applications and it will make it easier to write out some of the code that you would like to use online. There are quite a few things that you are able to do when creating your applications with Java, and we will take some time to learn how these work in this guidebook.

There are a number of benefits that you can enjoy when working with the Java programming language. While there are some other languages that you can use when coding, none are going to work quite the same as you would find with Java. Some of these great benefits include:

- The language is portable: since Java is able to work with a wide variety of systems and browsers, it is considered a portable language. Developers are able to write the code out and then know that it will work on any browser, and for any user, who has the Java plug-ins.
- Safe and secure: sometimes JavaScript is criticized as not being the most secure and with all the hackers and other bugs that can come up working online, most users want something that is secure. Java is a secure language with some high

level encryption that will ensure that all information shared is safe.

- Easy to connect: The Java program is designed with some simple phrases so that any developer is able to get onto it quickly. Not only is the language easy to read, but you will find that hooking up with the Java language and getting it to work for the system can be faster than ever.
- High success rate: as long as you are using the program the right way and writing out your codes correctly, you shouldn't have any issues when it comes to working with Java. Java is known for being reliant and the compiler is great for reading through the code with little issues and making sure that your codes are executed properly.
- Powerful environments: the environments that are with Java are some of the best out there. NetBeans and Eclipse are easy to work with and have lots of extra power to get the code done. They help you to complete your code, but also include debugging capabilities to help you make your code work better than ever before.
- Open source libraries: these open source libraries will ensure that Java can be used anywhere that you would like. Google, Apache, and some other organizations have contributed to these libraries, which has made the Java development easy, cost effective, and faster.
- A big community: sometimes you are going to have questions or will need someone to help out with a particular code. The community inside of Java is large and it has a lot of support for those who are getting started with Java. Don't be afraid to go online and see some of the support that you can get from this language any time that you have questions.
- Documentation support: Javadocs is an additional tool you should try out. It will tell you a lot of things about the API with Java. Javadoc has made learning Java easier and can provide you with some references that are needed when coding with this.
- The platform is independent: with some codes, you will only be able to use the code one time, or use it on one kind of browser. But with Java, you are able to write the code once and get it to work anywhere that you would like. This makes it easy to write a program that anyone will enjoy.
- Free: nothing is better than picking out a programming language that is free to use. You will be able to download Java and its environment for free, allowing

you to get started with your own coding without having to deal with a lot of expenses. When you want to learn a bit about coding, going with the free option with Java is one of the best options for you.

- Easy for beginners to learn: as a beginner, you may be worried about how to learn coding for the first time. Perhaps you have listened to others talk about coding and feel that it is too hard for you to learn how to do. In reality, this is one of the easiest languages to learn when it comes to web pages and making some of your own codes for your website. It only takes a bit of time to learn how to use Java and even if you have no experience with the language, it is something that you can catch on to quickly.

When it comes to picking out a coding language that is going to work for your needs, will work on various platforms for many users, and can make some cool programs in no time, the Java coding language is the right one for you.

Downloading Java

Once you have decided that it is time to work with the Java language, it is time to download all the parts that you need. The Java program is free so you should be able to download it on your system without too many problems. In fact, it is common for this to already be on your computer so double check before you try and download it. Even if Java is on your computer, you will still need to add on a few more options to get it to work for writing codes. Here are some of the simple steps that you can take to download Java on your computer and get it to work right for you:

Install the JDK: this is one of the tools that you should use when it comes to developing your application on Java. You can install it at the same time that you download Java so it shouldn't be too hard to work on.

Choose the IDE: one of the best ones that you can use with Java is NetBeans. It is free and will download in just a few minutes. You will not be able to write out any code inside of Java if you don't have the right environment and you are able to pick any of them you want to use, but NetBeans is one of the best and easiest to use.

When all of these have time to download on your system, you will be able to start working on your program. These will just take a few minutes and they don't take up a lot of space inside your computer so you won't need to worry about the computer slowing down. Once your IDE comes up and the Java coding is there, you will be able to work

and writing some of your first codes.

Writing Your First Code

Once you have had time to download Java and your environment, it is time to give it a try by writing out your first codes. In order to successfully do this, you need to learn some of the commands that are common with Java. These commands can be typed into your text editor and will then tell the program what you want to have done.

Writing code can be simple inside of Java, you just need to get some experience with working inside of the system and with your text editor. To get started with your first code inside of Java, take the following steps.

Step1: to start writing a program in Java, you will first need to set up your work environment. Open up NetBeans or whichever environment you chose for working in Java and get it ready to use.

Step 2: once the environment is up, open up your text editor that you want to use. Notepad is a good selection if you have a Windows computer, but anything that is similar will work out great.

Now we are going to create the Hello World program. This is an easy program to use on any programming language that you want to work with because you will get the hang of how the syntax works and you will get the words “Hello World” to show up on your screen.

To start this, just go to your text editor, click on new file, and then save this document as “HelloWorld.java”. HelloWorld is going to be the name of your class, so keep this in mind since the class name needs to be the same as the file name.

One the file is created, it is time to declare the class and the main method. The main method will be the same in terms of method declaration no matter what kind of program you are creating inside of your Java program. At this point, you should have the following syntax:

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

```
}
```

Write this part out into your text editor. Next, you will need to write out the part of the code that will tell your text editor what you would like to have printed out. For this point, you would need to type out:

```
System.out.println("Hello World.");
```

This is going to tell the system that you want to print out the phrase “Hello World” onto the screen. You can try this out and change up the message you want to use inside of the program based on what you want to program to use. With this part, there are a few things that help to make it get done the proper way so let’s take a look at how these work.

- System: this part is going to alert the system that it needs to do something.
- Out: this is going to tell the system that you are creating an output that you want it to print on the screen.
- Println: this part will stand for print line in the system. You are basically telling the system that you want to print out the statement that comes after this part.
- Parentheses: there are some parentheses that are found around the “Hello World” part. These means that the code in front of it is taking in a parameter, or the string of Hello World.

So basically this code is working to alert the system that you need it to take some actions, that you want to create an output and that you want to print out the line “Hello World” onto your screen.

Before continuing, there are a few rules that you should follow when making these codes inside of Java. First, it is a good idea to add a semicolon at the end of the lines. This is a good programming practice and helps the text editor to print things off properly. Java is also a case sensitive language so pay attention to whether you are using upper case or lower case letters when you are writing out class names, variable names, and method names. And finally, any blocks of code that are specific to a certain loop or method will be encased with the curly brackets.

So let’s put this code all together so you can place it into your text editor in the right places.

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

```
        System.out.println("Hello World.");  
    }  
}
```

Once this is written into the code, it is time to save your files and then open up a command prompt, also known as a terminal, in order to compile the program. You can navigate to the folder where the HelloWorld.java is located. Type in the words `javac HelloWorld.java`. This is going to tell your compiler that you want to compile your HelloWorld.java.

If there happen to be any errors in your code, the compiler will be able to tell you what may be wrong with your code. Otherwise, the compiler is not going to show any messages. You can then look at the directory where you store the HelloWorld.java and inside should be the HelloWorld class that we designed. This is the file that is used when you want to run your program.

Now that we have written a program, you may want to give it a try and see if this will run. Open up the terminal or the command prompt and type out `java HelloWorld`. This is going to tell Java that you wish to run your HelloWorld class. If everything has gone into the program properly, the statement "Hello World" will show up on the console and you have written your first code in Java!

Expanding the Hello World Program

Working with the Hello World program was pretty simple and has given you some experience with how this program is going to work. Here we are going to take the Hello World program and extend it out a bit. In the Hello World program, we printed out a string for the user to see, but now we are going to extend out the program so that the users are able to place in their name and then it will greet them by name.

First, we will import in the scanner class. In Java, there are a few libraries that you can access, but you first need to import them. One of the libraries that we need and which holds the Scanner object is the `java.util`. to import this Scanner class, use the following code:

```
import java.util.Scanner;
```

this tells the program that you wish to use your Scanner object, which is inside the

java.util package. Inside of the method, we will instantiate a new instance with this Scanner object. To use this Scanner class, we just need to create a new Scanner object that we are able to populate and use the methods of. To get these Scanner objects to work for us, we would need to use the following code:

```
Scanner userInputScanner = new Scanner(System.in);
```

So what does this code mean and what is it going to tell the computer to do? Here is an explanation of how the different parts move:

- `userInputScanner`: this is the name of the Scanner object we are using. Note that this is written in what is known as the camel case; which is the conventional way of writing out your variables in Java.
- the new operator will help to create a brand new instance of your object. For this instance, we created a new instance in the Scanner object with the new `Scanner(System.in)` part.
- This new Scanner object is going to take in a parameter that will tell the object what they need scanned. The `System.in` is going to work for this and it tells the program that you want to scan in the input from the system, which is basically going to be the name of the user.

Now we can work on prompting the user for their input. We need to ask the user for the input so that the user has an idea that they are supposed to type something into their console. Otherwise, the program will just sit still because nothing is entered. You can accomplish this with the following code:

```
System.out.print("What's your name?");
```

The next part that you should do is ask the scanner object to take in the information that the user types and then have it store that information as a new variable. The Scanner is responsible for taking in the data that your user is typing, which should be their name at this point. To do this, use the following code:

```
String userInputName = userInputScanner.nextLine();
```

This should tell the Scanner object that you want it to read what the user inputs into the system and use that as the variable for your next part. It can now be used as the greeting that shows back to the user. Since you have the name of the user, you can write it out so that the program mentions the name while leaving another message. The next step is to

write out the following code:

```
System.out.println("hello" + userInputName + "!");
```

At this point, we have been separating out the code to the different steps and discussing it, but let's put it all together to help you see how this code should be written out:

```
import java.util.Scanner;

public class HelloWorld (

    public static void main(String[] args) {

        Scanner userInputScanner = new Scanner(System.in);

        System.out.print("What's your name?");

        String userInputName = userInputScanner.nextLine();

        System.out.println("Hello" + userInputName + "!");
```

Once this code is in the system, you will be able to compile the program and run it. Go into your command prompt or the terminal window and then run the same commands that we did with the HelloWorld.java from the last section. You can compile this program by typing in `javac HelloWorld.java`.

When you type in the HelloWorld.java, your program should ask "What's your Name?" You can go through and type in your name. Let's type in the name Jane. Once you press enter on this name, the program will print out Hello Jane and then the program is done.

While this part may take a few more lines to complete, it is still pretty basic. It took the original skills you learned with your first part and expanded it out into a code that has a few extra parts that come with it. Despite needing a few more parts, you will find that this is a basic code that adds in a bit more personalization to the codes you make.

And now you are set to start with the Java program. You already know how to do two basic codes inside of this program and as you move through the other parts, you will be able to write out some more codes to get the hang of how the text editors work inside this program.

Summary

Working in the Java coding language can be a great challenge. It is a fun language to learn how to use and it works on many different platforms so your user will be able to find it no matter where they are. Some of the things that we discussed in this chapter include:

- What the Java programming language is and why you should choose it.

- The important of setting up an environment to get Java to work properly.

- Some of the benefits of using Java for your programming needs.

Chapter 2: Working With Variables in the Java Language

Inside the Java language, a variable is basically a storage unit that will hold onto certain data types. The programmer will be in charge of naming the variable and can also decide what kind of data is stored inside of each variable. Your variables are easy to work with and you will be able to access them or change them any time that you wish. There are a lot of things that you can do with variables including taking information out of them, copying the information or moving it over to a new variable.

Learning how to work with the variables, along with the different types of variables that are found inside this language, and about the different types of data that will go into these variables will help to make some of the other aspects of this language, such as classes, conditional statements, and functions, work better.

Let's take some time to look at how variables work inside of Java and what they mean for your coding needs.

Types of Variables

Variables are one of the first important part of the Java language that we will look at. These variables are like little containers inside of the code that will hold on to all of the different data types that you will create inside our code. As the programmer, you are going to have a lot of freedom in what you are able to do with these variables, including how to name the variable, how to access them, and even how you would like to change them.

First we should look at the variable types that are unique to Java. All the variables inside this system are going to have a specific type that will determine how the memory is designed as well as the size of the variable. There are three types of variables that are defined in Java including:

Instance Variables

The first type of variable is the instance variable. These are declared inside a class, but will be accessed outside of any block, constructor, or method. These variables are

created with the “new” keyword and will be destroyed once their object is destroyed. They are also visible to all blocks, constructors, and methods inside the class it is declared. You can also give the instance variables the “public” keyword if you would like it to be accessible by the subclass within a class, but it is recommended that you set the accessibility to “private” whenever possible. You should also notice that all of your instance variables will have a default value, unlike the local variables.

An example of an instance variable includes the following:

```
public class Test
{
    Public int num; // integer named num

    Public void dog()
    {
        Num = 3;
    }
}
```

With this example, we declared your integer “num” outside of the methods and you will be able to access it at any time within all of the methods that are inside the “Test” class.

Static Variables

These variables will need the “static” keyword to be declared and they can be declared outside of any block, constructor, or method. If your variable is static, only one instance of this variable is going to exist; it doesn’t matter how many instances of the object are called. For the most part, the static variables will not be used unless you are working with constant values, which are basically a variable that doesn’t change.

A good example of the static or class variable include:

```
Public class Test
{
    Public static final int num = 3; //integer named num

    Public void dog()
```

```
{  
System.out.println(num)  
}  
}
```

With this example, we declared the int called “num” and then went through and set it as public, static, and then also final. This basically means that you can access the int known as number from within your subclasses as long as you just do it once and the value never changes.

Local Variables

Your local variable is only going to be declared within a constructor, block, or method and they are only created when one of those are created as well. As soon as the method ends, the local variable will be destroyed. These variables are only found in one place so if you try to call it up from a location outside of where it is created, you are not going to see much success and they are not visible outside the area they are called either.

A good example of how a local variable will be called includes:

```
Public void cat(){ //method named cat  
  
Int x = 0; //int with value of 0  
  
}
```

With this example, a variable that is known as “x” is called inside the method of “cat”. This particular variable is only going to exist inside the context of cat, and you will not be able to use any other method in order to access or call this particular variable.

These three types of variables are going to help to set you up to writing the best code that you need. Whether you would like to call the variable up from any class or method that you want in the system or you would like to keep with the local variables so that your variable only shows up in the method it needs, the variables will prove very valuable with your efforts.

Data Types

Now that you know a bit more about the variables, it is time to look at the different data types that they are able to hold. Java has a many data types that are built in to the

program and which are used to store the different types of data that are predefined. In Java, these are known as primitive data types and they are the most basic of the data types in this language. It is also possible to create some of your own data types, but we will just take a look at some of the types of data for now including:

Byte

Short

Int

Boolean

Char

String

All of these will make it easier to write out the code you want inside of Java. The byte is one of the most basic of them that allow you to get started while integers are nice when you want to work with numbers, Boolean values are basically true and false options, and a string is like the statement inside your code. Each of them have their own job and will help you to get some of the results you would like out of the code.

Declaring Your Variable

When you are ready to declare your variable inside a code, there will be three parts; the stored value, the name of the variable, and the data type. First, when naming a variable, there are some rules that you should follow. Your variable name can be as long or as short as you would like, but it must start with either a letter, an underscore, or a dollar sign or you will end up with an error with the code. Once the first character is determined, you can use any letter, number, underscore, or dollar sign that you would like to name the rest of the variable. Often if you using more than one word with naming your variable, you would start the first word as lower case and then all the subsequent words would start with an upper case letter.

Using Variables

Once the variable is declared, it is time to read or change the value. Keep in mind that once you declare the variable, you will only be able to reference it by this name and you will only need to declare your data type at the time that you declare the variable. An example of how the variable would look inside your code includes:

```
int a: // no value
```

```
int b = 1; // value of one
```

```
int c = 2: // value of two
```

```
a = b+c; // sets a to the value of b+c
```

in this example, you changed up the value of a so that it equaled the sum of b and c when they are added together. When you started, you had a with no value, but when all the variables were used together inside the code, you're a is going to have a value of 3. You can do many codes that are similar to this to show how the variables will work inside the code.

Assignment

As a quick exercise, let's take some of the information that we have learned about with variables and create a calculator that will take care of adding together numbers for us. The first thing we are going to do is to declare three variables; one to store your value in, one that is going to represent which number we are adding first, and then one to represent the second number you want to add. We are going to declare these variables as double so that it is easier to add in decimal points later on. Our code is going to look something like this:

```
double a = 0; // stores value of addition
```

```
double b = 3.55; // first number to add
```

```
double c = 52.6; // second number to add
```

Now that these are all declared, you will be able to type in their variable to get the right answer. So let's set the value of a so that it is equal to when b and c are added together, and then have the system print out what the value of a ends up being. Type in the following to get this to work on your environment:

```
a = b + c;
```

```
System.out.println(a);
```

When this program is running, you will get the answer 56.15. With this simple code, you have worked on creating your first calculator. Try out a few different numbers and get

used to working around with this option by changing some of the data types, adding in some more numbers, and learning how it feels to write out some of these codes with variables in them.

Other Data Types in Java

In addition to working with variables inside your code, there are other types of data that are helpful to making sure that the code works the way that you would like. These can be used in different ways depending on how you want the program to work and how the compiler will read it. Let's look at some of the other data types inside Java and learn how they work inside the codes you want to write.

Characters

This is the basic type of data that you can use inside the code. They are the words that you can use in order to create the code and can sometimes even be the commands that you send out. Some examples of characters that you can use include:

- If—this is going to work on a true and false basis. You can also use the if...else and the elif statements based on what you want to get done with your block of code.
- While—this is often used for true conditions or to execute out blocks of code.
- For—this is used for blocks that are attached.
- Try—this is handled by except and will make it easier to raise exceptions for your code blocks.
- With—this will use your block managers and can be used to help out with opening and closing files.
- Def—this means that your function or your method will be defined in the code.
- Class—this is going to be used in object oriented programming and will execute the blocks of code.
- Print—you can use this when you want the phrase to be printed on the screen. For example, after someone puts in their age or a number, you can print out some kind of phrase to tell them they are right or wrong.
- Import—if you are trying to import some modules from your other functions, the import can be nice.
- Yield—this is going to be used as an operator and it is used to generate function values.

- Assert—this is often used to check out the statements that you want to have applied.
- Pass—this will allow you to create a code block that is empty.

One thing that you should remember with the characters is that they can't be changed into variables when they are used as characters. They are reserved commands in Java and they tell your text editor what you want to happen during code execution.

Statements that take more than one line

For the most part, your statements will take up just one line because they are short enough. But what happens when you want to write out a large statement after the command. Will the Java program be able to recognize that it is still part of the same statement or will it try to read this as a new line of commands and send out an error message?

If you are writing a statement with your command that takes up more than just the one line, you should add a back-slash to the beginning of that statement. In addition, the brackets will work too. Choose one or the other of these, rather than both, to state where the statement is going. Once the statement is done, move on with a space to a new line.

Comments

Here we are going to take some time to talk about comments and how they work in our code. Looking through some of the codes in this book, you may notice there are parts that have a hashtag before them and then a small sentence. These are examples of comments in the code and they are an important piece of the puzzle.

Comments are not crucial to the code (and if the compiler read them it would get confused and result in an error), but they can help other programmers who are reading the code to understand what is going on inside the code. If you want to explain yourself in the code, or tell the other programmers what needs to be entered in this part of the code, a comment can be useful.

Making comments in Java is a simple process. you will just need to add a hashtag before the comment, write out the statement that you want there, and then skip to the next line to continue writing the code. The compiler recognizes the hashtag and will just skip right over this part of the code so it will never show up on the screen for the user. You can use as many of these comments in your code as you would like, but keep it to a minimum to avoid issues with readability in your Java code.

Boolean expressions

You will find that the Boolean expressions will be helpful when we get to the conditional statements later on. The Boolean expressions are going to be based on true or false. For example, if you only want someone who enters their age to be above 21, you would have a true show up when the person enters any number above 21, but a false any time they enter a number that is 20 or below. If the true expression comes up, you will have a certain statement come up, such as “Congratulations, you qualify to participate.”

Now, depending on which way you do this, there are a few things that can happen. In the simplest of forms, you will have the program just end if the statement ends up false, or the person enters an age that is 20 or below. But you can go through and make some changes so that instead of having the program just shut down when a false occurs, another message comes up. For example, when the person enters a number that is 20 or below, you can make it so that the program will show a statement such as “Sorry, you are not able to participate at this time.”

Remember that you are the one who is in charge when it comes to the code that you are writing. There are a lot of expressions that you are able to use that fit into the Boolean expression, such as the “if”, “if...else” and “elif” statements that allow you to create the exact code that you are looking for no matter what program you are working with. You will be able to determine a few different true and false statements that will work the best with the code that you are making.

Integer numbers and floating points

While working on the Java code, you can expect that there are times that integer numbers and floating point numbers will be used to store any numerical values inside the code. They can also be grouped together. The types of numerical points that are recognized inside of Java include:

- Int (signed integers)
- Complex (complex numbers)
- Float (floating real point values)
- Long (long integers; these could be either octal or hexadecimal)

Operators

Operators are a useful type of data that you will be able to use in much of your code. They are simple, but they can open up a lot of possibilities for what you are able to do inside of the code. In Java, there are a number of operators that are available including:

- Arithmetic: you can probably guess how the arithmetic operators work. They will allow you to do mathematical equations with two or more numerical values within your code. You are able to add, subtract, multiply and divide your values along with a few other options. You will also be able to do multiple signs inside the same code, just remember to follow the rules of operation to ensure that you are taking care of these numbers correctly.
- Identity: the identity operator is helpful when you want to compare the memory value of two objects, especially when they are located in two different places.
- Assignment: next are the assignment operators. These are the ones that will assign a value to the variable you are using. If you would like to put a letter, statement, or even number to one of our variables, you would use one of the assignment operators.
- Bitwise: if you would like to go through the operator bit by bit, you would want to use the bitwise operator. This option will go deep into the code, using the standard 0 and 1 to look through the code rather than looking at the words and numbers that are actually on the screen.

- Membership: the membership operators are going to be used in your code to see if there is some kind of coherence, also known as a membership, within the sequence that you are working with. They will often work with the Is and the No Is to determine if there is a connection.
- Comparison and relational: comparison and relational operators go together. You will be able to take two parts of the statement and compare them to find out if they are the same or different. Often you are going to use them to see if your statements are true and if they are, a certain task will occur in the program. If they are false, you may have to change up the program to make sure that it is providing the right answers in this case as well.

All of these data types are important because they help you to work with the variables and get some of the answers that you want out of the system. They can be as simple as working with basic characters to write out your statements or they can be a bit more complex such as operators and integer values. All of them are important and help you to properly write out the code you want in Java.

Summary

Variables are basically the storage containers for data inside of the Java programming language. There are three basic types, instance, static or constant, and local variables, that will work to make writing inside of Java much easier to learn.

In addition to being able to work with many types of variables, there are also several data types that you can place into these variables. These data types are powerful and can teach your program how to behave in the manner that you would like.

Variables are simple parts of your program that are easy to manipulate and make your own code. As we showed inside this chapter, it takes just a few lines of code to make your own calculator using the right data types as well as storing them properly inside your variables.

Chapter 3: Working with Conditional Statements

While working in Java, you may come across times when you would need the program to make decisions. These decisions would be based on a combination of the conditions that you set and the input that the user gives to you. Depending on the conditional statement that you choose, the program will determine if the input is considered true or false based on the conditions that you set.

In technical language, these are called the conditional statements, but we can also call them the if, the if...else, or the switch statements. Each of these will work slightly differently. For example, the if statement will only have one option and is looking for the true input. If your user places in a true answer, the program will execute, usually with a statement that you placed in. With the if...else statement, you can place two conditions inside. If the input from the user is true, your user will see one answer, but if their input is false, your user can see a different answer.

Both of these are needed to help you to create some power inside of your code and gives you some control over the response of your program based on the input that the user places in. Let's take some time to look over these conditional statements and see what all they can do.

The If Statements

When working with the if statement, we are talking about a true and false method. This statement is going to decide whether the input of the user is considered true or false and then it will give the response that is needed based on the conditions you set. You will be able to choose the conditions that are true based on your program.

You can determine the statements that you would like to come out when the user puts in an answer that is considered true. But with the if statement, when the user puts in an answer that is false, the computer isn't going to show up anything. This is part of the limitations with the if statement; nothing is going to come up when the answer is false, such as when the user puts in the wrong answer to $2 + 2$. A blank screen doesn't look that good when your user is on the program.

You may choose to just have a message come up if the person is older than 18. You would need to use the ≥ 18 sign in the code and then add in the message that you would like. For example, if the person placed their age as 35, you could have the program provide the statement “Congratulations! You are able to participate!” and then it would go on to the next part of your code. On the other hand, if your user places their age as 16, the program would either end or it would move on to the next thing that you want to show up in your program.

The If...Else Statement

When you are working with the if statements, you are going to have some limitations to what the code is able to do. When the user puts in the right answer, they are going to get the statement or message that you list out. But what happens when they make a wrong answer, or at least one that is considered false based on the conditions that you put inside the code?

There are going to be times when your user will put in an answer that is considered false based on your conditions. This doesn't mean that the user is wrong. But if you set up a program that only allows for ages that are 21 and over, the only time that your user will get a response with the if statement is when they are over the age of 21. When they answer that they are 18 or some other age that is under 21, you are going to get a blank screen.

This is not going to look very professional for your code. The user may assume that something is wrong with your program since nothing is showing up. And this is why learning about the if...else statements is so important. These are meant to provide one answer if your conditions are met, and then a second answer if the conditions are not met.

So with the example above, let's say that your program asks the user what their age is, and you are setting it up so that the answer is true only if the user is 21 or above. If the user puts in that their age is something like 25, the system will see that this is a true answer and will display your first statement. You can choose what statement you would like to show up here, such as “Congratulations, you are allowed to enter the site.”

Then, instead of turning the program off, when the user places in an age that is below 21, such as 18, the program will see that this is a false answer. It will skip the first statement in your code and move on to the second statement that you put in. This message could be something like “Sorry, you must be 21 to enter this site.”

You are able to add in as many of these if...else statements to your code as you would like. Sometimes it is good to separate out things into different categories. For example, say you are asking your user what their favorite animal is between cat, dog, horse, or fish. If the person says that they like cat, your program would see that answer as meeting the conditions and being true and would display the function that you have there. But if they state that fish is their favorite answer, the program would find that the three answers above were all false, and it would keep going down the line until it ran into the one that is true.

This is a pretty simple way that you will be able to add in some more options to the program. The user will be able to input the answers that they would like and you can set up the program to respond based on the answer that the user gives. While the if statements are good ones to get started with if you are not used to programming, in most cases you will want to set up the if...else statement to ensure that the program will continue on regardless of the answer that the user gives.

The Switch Statements

Another conditional statement that you may choose to use inside of your code is the switch statement. This statement is going to make it easier to test out the variables inside of your code to see the equality of them against a list of known values. A singular value in this option is going to be called a case and the variable that is switched on is going to be checked for each of the existing cases. An example of doing an enhanced loop that has a switch statement inside of it will include:

```
switch(expression){  
    case value:
```

```
//Statements
Break; //optional
Case value :
    //Statements
    Break; //optional
    //You can have any number of case statements.
    Default : //Optional
    //Statements
}
```

Summary

Conditional statements can make it easier for the program to make decisions based on the conditions that you set out and the input from the user.

The if statements are the most basic type of conditional statements. They will be based on the true and false idea, but they only work when the input from the user is true.

The if...else statement is meant to help fix some of the shortcomings that come from the if statement. When the user places the right answer into the program, you will get one statement, but another statement when they put in an answer that is seen as false.

Switch statements will help you to create comparisons to see if a set of variables and values are equal to the set that are inside the statement.

Chapter 4: Iteration Statements and Looping

Statements in Java

When working with Java, you may come into situations where you must loop through a large amount of numbers in the program. If we use some of the tips we have learned so far to loop through a hundred numbers, or perhaps more, we would need to use one hundred of the “if” statements from above. This is not really practical; who wants to spend all that time writing out a hundred if statements. It is messy, takes up a lot of time, makes the code harder to read, and is just not necessary.

Luckily, Java is set up to avoid some of the issues with having to write out that many if statements thanks to using loops. There are three loop types that work in Java and deciding on when depends on what you would like the code to do. This chapter will look at the three types of loops you can use and how each of them work.

The While Loop

The first loop to look at is the while loop. This is a control structure that allows you to repeat the task as many times as you choose. Instead of having to do the if statements to count for example, you could set up the while loop to repeat the same task a hundred times. It is more efficient in space and the time it takes you to complete. The syntax to use for a while loop includes:

```
while (expression) {  
  
    // insert code  
  
}
```

This is going to work like the if statements we discussed before, but this one is set up to continue looping through the code for the entire time the expression is true. Once the expression becomes false, the code will stop looping. It is also possible for the loop to continue until the user ends it if all the conditions keep coming up true. Let’s take a look at an example of how the while loop would look with some information inside:

```
int x = 10;
```

```
while (x>0){  
    System.out.println(x);  
    x--;  
}
```

For this code, the value of X is going to print out, and then it will continuously subtract one from this value until you end up with x being greater than or less than zero. If you ran this code, you are going to get an output of all the numbers from one to ten. When your x value is zero, the expression is going to return a false value, and the loop will finally end.

Watch out for infinite loops in your code. This is basically going to freeze up the program because nothing is going to end up making the loop false. It usually happens when there is some error with your logic and so the system sees that all the answers are true, therefor repeating the program over and over again.

The Do While Loop

The do while loop is going to work in a similar manner as the while loop, but there is one difference. The “do”, or the part that happens when your statement is true, is going to be called before you ask the condition. This means that the do while loop is always going to run through at least one time. Rather than waiting to see if the statement is true, the do while loop will complete one loop, and then check to see if the condition is true. If it is true, it runs a second loop and so on. Once the do while loop finds that the condition is false, it will stop running.

A good example of the do while loop includes:

```
int x = 10;  
  
do{  
    System.out.println(x);  
    x--;  
    while(x>0);
```

in this example, the program logic from your while look is transferred directly to the while loop. The only difference in this syntax compared to the while loop is that the

logic that runs when your statements are true is going to be above the while loop, rather than below it. This output is going to be the same as with the while loop, but the following example will make some differences:

```
int x = 0;

do{

System.out.println(x)

x--;

}

while(x>0);
```

For this example, the output would be different compared to the while loop. If you wrote this out as just a while loop, the code would never run because it would end up being false the first round. But, with the do while loop, the loop will run once, see that the statement is false, and then stop. So if you would like to make sure that the loop runs at least one time, you will need to use the syntax for the do while loop.

For Loops

For loops will help you to do increments and loops through ranges of values inside of your statement. The syntax you can use with the for loop is as follows:

```
for(initialization; termination; increment){

//statement

}
```

In this code, the initialization is going to initialize the counter variable that you would like to use. Termination will state the expression that the system should evaluate to see if it is true or false. And the increment is going to be the increments that are initialized in the counter variable. An example of how this would work with coding includes:

```
for(int x = 0; x < 10; x++){

System.out.println(x);

}
```

For this example, we start by bringing up the for loop. We then create a new int that is

called “x” and set it to be zero. Next we state when this int will terminate, saying that the loop should continue as long as x stays less than ten. We will then set the increment of increase for the x value to be by one.

The initialization is only going to occur once, right when you call the for loop. After this is initialized, it will run at least once. Once one loop is done, it is going to call the next increment value and then check to see if the program should terminate. If it doesn't get terminated, the increment value would be called again. This would continue on until the system determines that the statement is false and it is time to stop. With the code above, the loop will repeat with all the numbers from zero to nine, but will terminate when it reaches 10.

The loop statements can help to simplify your code while getting the right information to show up in the compiler. For example, the regular loop statement will help to run the program after reading the statements and determining whether the answer is true or false and then the do while loop would run the program first before checking the statements and determining if they should repeat again. These will save you a lot of time and effort and can provide the same power that you want inside your coding.

Summary

Loop statements can save a lot of time and effort by telling the computer to keep going through the loop until the statement comes back false.

The do while loop statement works slightly different compared to the regular loop statement in that the do while will run first, before checking to see if the conditions are met. This one is good if you want to make sure that the loop runs at least once.

The for loops are good for working with increments and ranges inside of your loops.

Chapter 5: What are Arrays and How Do They Work in Java?

There are times when you would like to combine together several of your variables inside of your code. You could choose to list them out in a row in the code, which can take up a lot of time and makes a mess with the code, or you can choose to work with some arrays to put them all together without all the hassle. This chapter is going to take a look at how arrays work in Java and how you can use them to create some powerful codes.

Arrays

In addition to using loop statements to make things easier in your coding, arrays are going to add to the code as well. Arrays are going to be data structures that will store a fixed number of variables that are of the same type. Rather than having to create many different integer variables in order to store all the number values in your code, you can create an integer array that would be able to hold all these numeric values. This makes it easier to manage your integers, and it keeps the code easier to read through.

Declaring your arrays will be similar to declaring a regular variable. The syntax that you should use to create these arrays include:

```
dataType arrayName[];
```

this is the way that you are able to declare the arrays that are added into Java. It is similar to using the C++ and C programming so people with that background will be able to use this familiar syntax as well.

The array that we just wrote out doesn't currently hold onto any data. We declared our array, but we didn't add in any values so it doesn't know how big it should be, the values that should be inside the array, or any other information to set up the array. Here is an example of what you can set up in order to set up the array properly:

```
int intArray[] = new int [5];
```

this example is going to declare an integer array that have five slots available inside of

it. This means that the array that is called the `intArray` will be able to hold five different numbers inside of it. Each of these numbers can be accessed and edited on their own using the following method:

```
int intArray[] = new int[5];
```

```
intArray[0] = 2;
```

```
intArray[1] = 3;
```

```
//...
```

```
System.out.println(intArray[0]);
```

This code is going to declare that the `intArray` has five slots for data. You are then able to access these slots by adding in an `int` value into the square brackets, right beside the name of the array. When you work with the `intArray[0]`, you are referencing the value at the index of zero for this `int` array.

When you create an array that has five values, you are also going through and creating five slots in it, which will be numbered 0 to 4. When you declared the array, it was already going to contain the value zero, but you changed it up to hold onto more values. You will be able to access each of these independently, just like you can with a normal variable, but it is easier to keep things organized inside the code rather than writing out each part.

Multidimensional Arrays

The arrays we discussed above were pretty simple, adding just one dimension and layer to them, but you can make your arrays more complex for your code. When you add in more dimensions, you are going to add in new arrays within your array. An example of how this would look includes:

```
int intArray[][] = new int [5][5];
```

This array is going to have two dimensions inside. The first five will state that you are declaring an array that has five index values inside. The second array means that for each of the five index values above, you will be able to access an additional five index values inside of them. If you would like to reach the information that is in the second

level, you must first list out the level it is contained in and then the name of where the information is in the second part. Below is an example of how you can access information in your arrays:

```
int[][] intArray = new int[5][5];  
  
intArray[0][0] = 1;  
  
intArray[0][1] = 1;  
  
intArray[0][2] = 1;  
  
intArray[0][3] = 1;  
  
intArray[0][4] = 1;  
  
// and so on until you get them all.
```

Combining Together Loops and Arrays

So we have spent some time talking about how to use loops and how to use arrays in your code, but now it is time to combine them together to get a better production with your code. Loops will allow you to quickly assign and retrieve the values for really large ranges, while arrays make it easier to create some large ranges of numbers as well. How efficient and nice would it be to combine them? Below we will have an example of how to write out the different kinds of loops, how to loop through your arrays, and even to edit your multidimensional array:

Example 1:

```
int intArray[] = new int[100];  
  
int x = 0;  
  
while (x < 100){  
  
    intArray[x] = x;  
  
    System.out.println(intArray[x]);  
  
    x++;  
  
}
```

This example, we declared our array and give it one hundred index values. When that was done, we declared our int that we wanted to increment. We told the while loop that it should continue to loop for as long as the value of x is less than the length of the intArray, which would be the number of indices the intArray has. Then we did increments of x going up by one for each iteration. This code allows you to quickly edit one hundred values and then prints them out onto the console.

Example 2:

```
int intArray[] = new int[100];  
  
int x = 0;  
  
do  
{  
    intArray[x] = x;  
    System.out.println(intArray[x]);  
    X++;  
}  
  
while(x < 100);
```

For this example, we will get some of the same results as the first one, except we designed the do while loop instead of using the while loop. The output is going to be the same, but you will make sure that the loop is going to work through at least once.

Example 3

```
int intArray[] = new int[100];  
  
for (int x = 0; x < 100; x++){  
    intArray [x] = x;  
    System.out.println(x);  
}
```

For this example, we are doing a few things similar to the other options, but the counter variable is going to be built into your loop. This means that you won't need to go

through and declare another variable, or tell the system to manually increment it for you. This is a more efficient method of working with your arrays and loops without having to write out as much code.

Example 4:

```
int[][] intArray = new int [100][100];  
  
for(int x = 0; x < 100; x++){  
  
for(int y = 0; y < 100; t++){  
  
intArray[x][y] = y;  
  
System.out.println(intArray[x][y]);  
  
}  
  
}
```

This example is one of the best ways to show how to work with a loop through your multidimensional array. You will have two loops working inside of each other and each of them will be responsible for a specific dimension in your array. For this one, the first time you get a loop, it will work with the $x = 0$ part and then it will loop through all the array indexes where the index starts with zero. Once all of these are called up, it will increment through x by one. It completes this process until all of the indexes are called.

Working with arrays and loops can help to simplify some of the complex codes you would like to write. For example, instead of writing out one hundred if statements, you could set up a loop to write out increments of 1 that go all the way up to 100. Arrays can help to store together some of your variables together, rather than having them listed out separately and making the code convoluted. You can even combine the arrays and the loops to create a complex code that will run through all the different combinations that you want inside the program.

Summary

At times, you will need to store certain variables together. Instead of writing them out individually inside of your code, you can put them together inside an array.

Arrays can have just one level, or you can add in more than one level to keep things together. We looked at just two dimensions, but you can add in as many as you would like.

Not only are arrays powerful by themselves in your code, but you can add them together with loops to keep the program repeating how you would like.

Chapter 6: Functions and Methods Inside Java

Functions and methods are important when you want to create a new program inside of Java. Methods are the way that you can communicate with the objects in your code. When you call up a method, we are asking for the object to carry out this task. We can basically say that the method will implement the behavior of your objects.

Each of your methods will need to give a name and you will need to define the input parameters as well as the return type to make it work properly. It is also a good idea to set the visibility of the method, including public, protected, or private. All of these parts will come into play when it is time to call your methods.

Working with the Functions

It is possible to define your own functions when using Java. When working in this coding language, you will find that everything that you do will be defined into a class, and that all the classes have methods. So, if you would like to define a unit that will do your own computations, you will need to define a class as well as the method that is inside your class.

Here we are going to look at an example of how this works. Remember that you should save this one in your text editor under F1.java

```
class F2 {  
    public int addone (int n) {  
        return n+1;  
    }  
}  
  
public class F1 {  
    public static void main(String[] arg) {  
        F2 x1 = new F2();  
        int m = x1.addone(4);  
    }  
}
```

```
System.out.println(m); // prints 5
    }
}
```

So at first this code may look a bit confusing and you are lost at what is going on with it. Here is some more information about what is going on in the code and what it all means for you:

- The codes that you create in Java are all going to be units of classes.
- Each file will be able to define as many classes as you would like. Many beginners assume that the file can only define one and then they take on more work than they should. The name of the file though must be the same to one of the classes that are in the file; doesn't matter which one, but it needs to be associated. For example, with the above code, we named the file F1.java and it corresponds with our F1 class.
- The main class in your file must have a method that is called "main". This is going to be the location inside the program that will begin the execution of the code. F1 was the main method in the example above.
- The other classes inside of your file need to work in a supporting role. This is why they are often called subroutines or functions.
- So, with the code above we created a main class that is F1 and then had the F2 be the supporting class. And then we defined the addone method inside of F2. We will call this method by creating an instance out of it, which will be the x1. Once this instance is created, we can call on the x1 method in order to do the computation of the code for us.

This is the same outline that you would use for many of the programs that you want to create in Java. This one is a bit more complicated than some of them, since there are two classes present at the same time, but the programs you create will still need the right functions in place to continue on.

These functions in Java are going to help you to bring up some of the codes that you need inside your language and will ensure that you are able to get the right methods pulled up when you need them most. Each of the functions will work a bit differently and have different information inside, but they will follow the same kind of outline as what is listed above.

Summary

Creating functions inside of your code can be done with just one method or with two.

You will need to name the file the same as one of the functions that is inside the code, usually the main one.

There can be additional functions inside the code, and they are going to be supporting functions.

Chapter 7: Working with Classes and Objects

Classes and objects are an important part of working in any programming language. They are used to help sort out some of the information that is inside the code and makes it easier to not only write the code but for other programmers to read the code and for your program to execute the code.

First, let's take a look at Java as an object oriented programming language. Some of the older versions of programming languages were not based on the OOP formula. These are the ones that often give non-programmers frights about getting into coding because they were more difficult to use. Things could easily get mixed up and not turn out the way that you want and the codes didn't make as much sense.

With the OOP languages like Java, things make a little more sense. The objects are going to be tied to physical objects in the real world and they are going to stay in place when you are done with them. For example, a ball in the Java code is connected to a ball that is in real life. This helps to make the code more readable and easy to use.

Classes and objects help to organize your coding and can keep things all in one place, rather than letting them float around. Let's spend some time looking at the objects and classes inside of Java and see how they can work for you.

What are Objects

First we are going to take some time to talk about objects that are in your code. Objects are the parts of the code that can be tied to physical things inside your world. They will have attributes, just like the real objects inside your world. For example, let's take a look at how a ball would work in this.

When you have a ball, you would find different attributes to give to it. You could say the ball was round, name the color, talk about if there was any texture to it, whether it was flat or filled with air, how it bounced and so on. These are all attributes of the object and will help to describe what it looks like and how it acts.

Inside your code, the objects are going to work the same way. They will have attributes to them to help describe them. As the programmer, you will be in charge of listing out the attributes that you want to go with your objects. You do need to keep in mind that the

attributes you use need to make sense to others reading the code.

For example, you could give the attribute round to the object ball, because it makes sense. But if you give the attribute triangle to it, this would not be allowed. The attribute needs to be something that others would place with the object, something that makes sense, otherwise it can't be considered one of the attributes of your object.

There are many ideas that can come up when working with the objects, and while some of them may be concrete, such as using ball above, others can be a little more abstract, as you may have noticed from some of the code already written in this book. The idea of the objects being connected to real items in our world is a better way to design the code, one that is going to ensure that this code stays put together and is easier for beginners to learn, is one of the benefits of using this language.

What are Classes

Classes are going to work with the objects. They are responsible for holding on to various objects to help keep the code organized. But, your classes are not just going to hold random objects in them; rather, they are responsible for holding together objects that belong together.

The objects that go into the class don't necessarily have to be exactly the same. You could have a class that included round balls and that is all that was allowed in there, but making a class for sports equipment could work as well. The latter would include not only round balls, but hockey sticks, pucks, helmets, uniforms, hurdles, footballs, and other things.

The items in the class need to fit under the same category, but they don't have to be the exact same as illustrated above. If it makes sense in the class for the objects to go together, then make sure to put them in the classes together. This can help you to bring up the right stuff when you need it in your program and it keeps the code more organized than before.

Working on the Framework for the Classes

There are a number of things that you should include in the framework of your class. These are going to help the program to work properly and will ensure that you are able to pull up the class whenever you need it to work. The things that you should include inside your framework includes:

- The program type. You will need to use the UNICODE declaration in order to describe the program type.
- Program information is going to be provided within the text. You should use the <HEAD> and </HEAD> symbols to surround this program information.
- The Java Program part is going to be described similarly, but the symbols are going to be <Java> and </Java>
- The title should be at the beginning and it is going to be found within the text <TITLE> and </TITLE>
- The text that you will see between the <H1> and </H1> are going to be the headings. If you want to have more than one heading show up, you would just need to change the heading number, such as having H2, H3, H4, and so on.
- The content of the program, or the information that you will see on the screen is going to go between the symbols <BODY> and </BODY>
- And if you would like to create a paragraph, but write out the text between <p> and </p>

Turning These Classes Into Pages

The ultimate goal of making classes in Java is to turn them into pages for your website. You want to be able to write out the new pages that you want to use, such as those for a website and blog, in order to make it work exactly how you need it to online. Below is an example of how you will be able to take your classes and get them to form into pages:

```
<!UNICODE Java>
```

```
<Java>
```

```
<body>
```

```
<h1> Welcome to the Web! </h1>
```

```
<p> My first page </p>
```

```
</body>
```

```
</Java>
```

This example will help you to write out the page that you want within your class. You can create the header to be whatever you would like, such as “Welcome to My Page” and then have the paragraph be as filled out as you would like; some pages may be basic with some information on them about what you are selling and others, such as when you are working on a blog page, can be more filled out to complete.

Creating Headings

Inside your class, you will need to make some good headings. These are going to help make the page look more attractive and ensures that others know what each part of the process is about. You can also add in some of the smaller subheadings if you would like to break up the information in the classes rather than having on long line of text.

Inside of Java, you will be able to range from <H1> to <H6> so use them as you would like. They are basic to put into your system. You can just use the following syntax, and then add in the paragraph information if you would like in between each one:

```
<h1> Look at my heading </h1>  
<h2> Today is a beautiful day </h2>  
<h3> Tomorrow I have no plans </h3>
```

You can keep on with this patter all the way up to the sixth heading. You will notice that when you execute these, they will show up with their own page so that it is easier for them to be in their own sections. You can add in some more words, such as paragraphs to show the information you would like.

A good way to look at this is with the idea of a blog article. You would have a heading at the beginning to explain what the information is about. You can then write a few rods before adding in a new heading and as many subheadings as you would like. In between each one, you would then write in the information that explains and fleshes out the subheading, making for a whole blog article.

Paragraphs

After you are done writing out some of the headings that you want inside of your code, it is time to add in some of the words that you want to work with. People are going to be on your web page in order to learn something, not just see a few headings. You can place anything that you want into the paragraph part of your document, but learning how to do this properly can make the process a bit easier.

To write out a paragraph in your coding, you would simple need to use the `<p>` and `</p>` symbols to indicate when you are starting and when you are finishing. You will be able to write as much in between these symbols as you would like. Sometimes it is just going to be a few lines or a short paragraph, and other times it is going to be longer, like a page. There are no limitations to how long your paragraphs can be, just make sure that they have the right symbols surrounding them.

Using Classes to Add in Images

To really make your web page stand out against the rest is to add in a few images to the mix. These enhance the articles that you are writing and if you want to sell something on these pages, it is good to have a collection of pictures to show all about the product. Adding in images can be simple in this language. You will simply need to use the `` tag around any pictures that you would like to add into the system.

Adding in Some Links

Do you have a product you want to sell and you need to take the user back to where the product page is? Do you have some information on another page that might interest your reader? If these are true for you, adding in a few links can help you to highlight your points. The `<a>` tag is the right one for doing this and it keeps things pretty simple. Here is an example of using the link tag inside your Java class:

```
<a href="https://google.com">See this link </a>
```

Adding in Some Quotes

Adding quotes can be a great way to spice up your page some more. If you have a topic you are talking about and find a good quote that really helps to showcase the point you

are trying to make, adding the quote into the page, rather than messing around with rephrasing it, can be the best option. It adds in some more value with your readers and is one of the best ways to showcase the point you are trying to make. To keep in good standing online, make sure that you provide a link (using the code and tag that we listed above) to show where the quote came from.

Here are a few the tags that you can use when you would like to add in some quotes to your class:

- `<q>` --this is going to tell the program that you are adding a short quotation.
- `<cite>` --this is going to be attributed to the title of the topic that is on the page.
- `<blockquote>` --this means that you have gone and quoted something that is going to come from outside sources.
- `<bdo>` --this states that the program should do a text dictation.
- `<address>` --this is the contact information quote
- `<abbr>` --this shows that there is an abbreviation or acronym that is used.

You can use these to write out any quote that you would like. You can use them to write out a short one-line quote, or even a quote that takes up a few pages because it is so long. It all depends on what you would like to have on your page. Just make sure that you link back to your original source to show honesty inside of your writing and to make sure that others are able to do more research on the topic if they choose.

Creating your classes the proper way inside of Java is important if you want to be able to create a page that others will want to come and visit. Going the wrong way and missing out on headings or on the different parts that go into the class will make it hard for you to get the page to look nice. Make sure to take down some notes about how you would like the class, or the page, to look when you are done and keep that in mind when you are working it out.

Summary

Objects are the physical parts of your code. They can be tied with real items in your world in order to make the code easier to handle along the way.

Classes are going to contain a variety of objects that have something in common together. Perhaps you are dealing with sea creatures, but each of them would be a different creature.

Since Java works with your online pages, it is important to learn the best way to write out your pages, including headers and paragraphs, to make the look amazing.

Conclusion

Thank you again for purchasing this book!

I hope this book was able to help you to learn some more about the Java programming language and how it can be used to help you to work on some of your very own codes. Even as a beginner, you can learn how to work within Java, it isn't that hard, and by the time you learn the concepts inside this guidebook, you will be ready to get to work and create the codes for your own program.

The next step is to download the Java program as well as the environment that you should use with this software and then get to work. We have provided lots of examples and exercises that you can try out in this book so open up the Java environment and text editor and try a few of them out. Don't be afraid to mix and match things and see how you would be able to use them for the codes that you have in mind. The more practice you get, the easier it is to make Java work the way that you want.

Finally, if you enjoyed this book, then I'd like to ask you for a favor, would you be kind enough to leave a review for this book on Amazon? It'd be greatly appreciated!

Thank you and good luck!