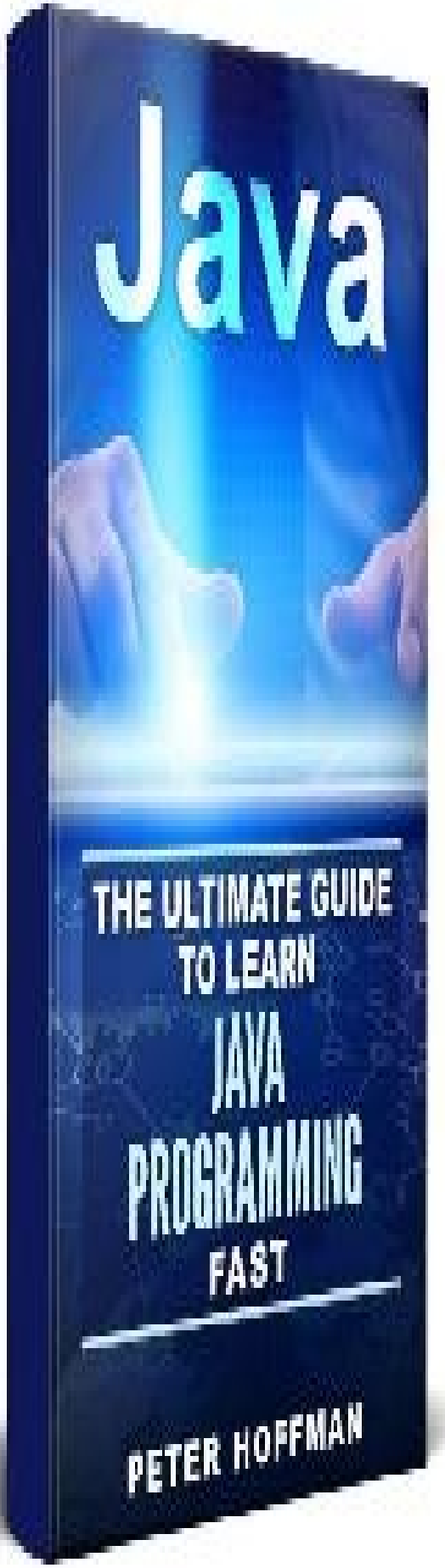
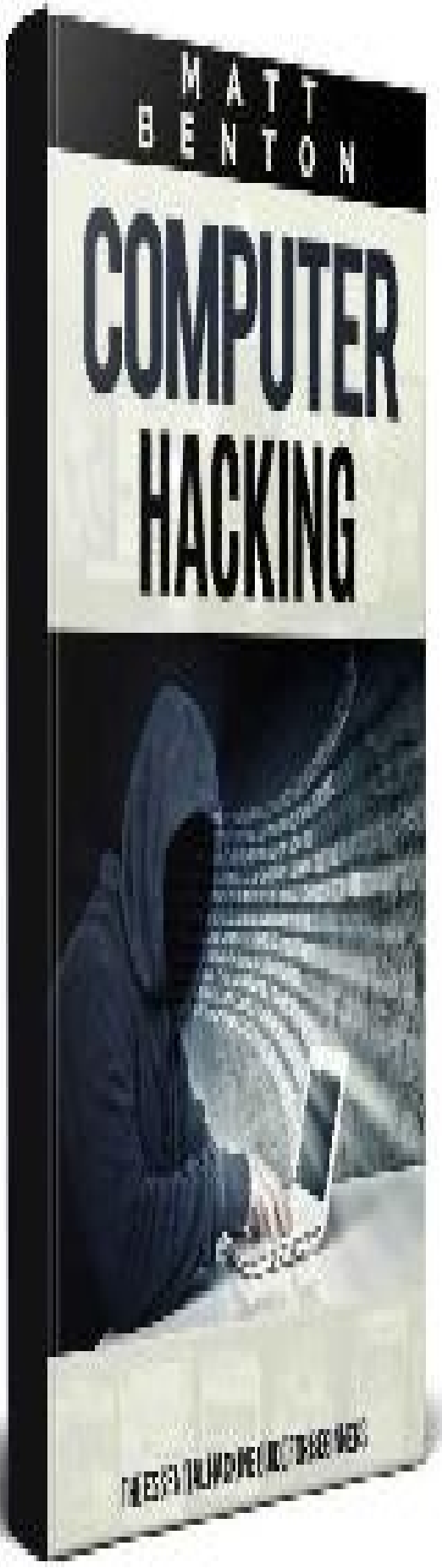
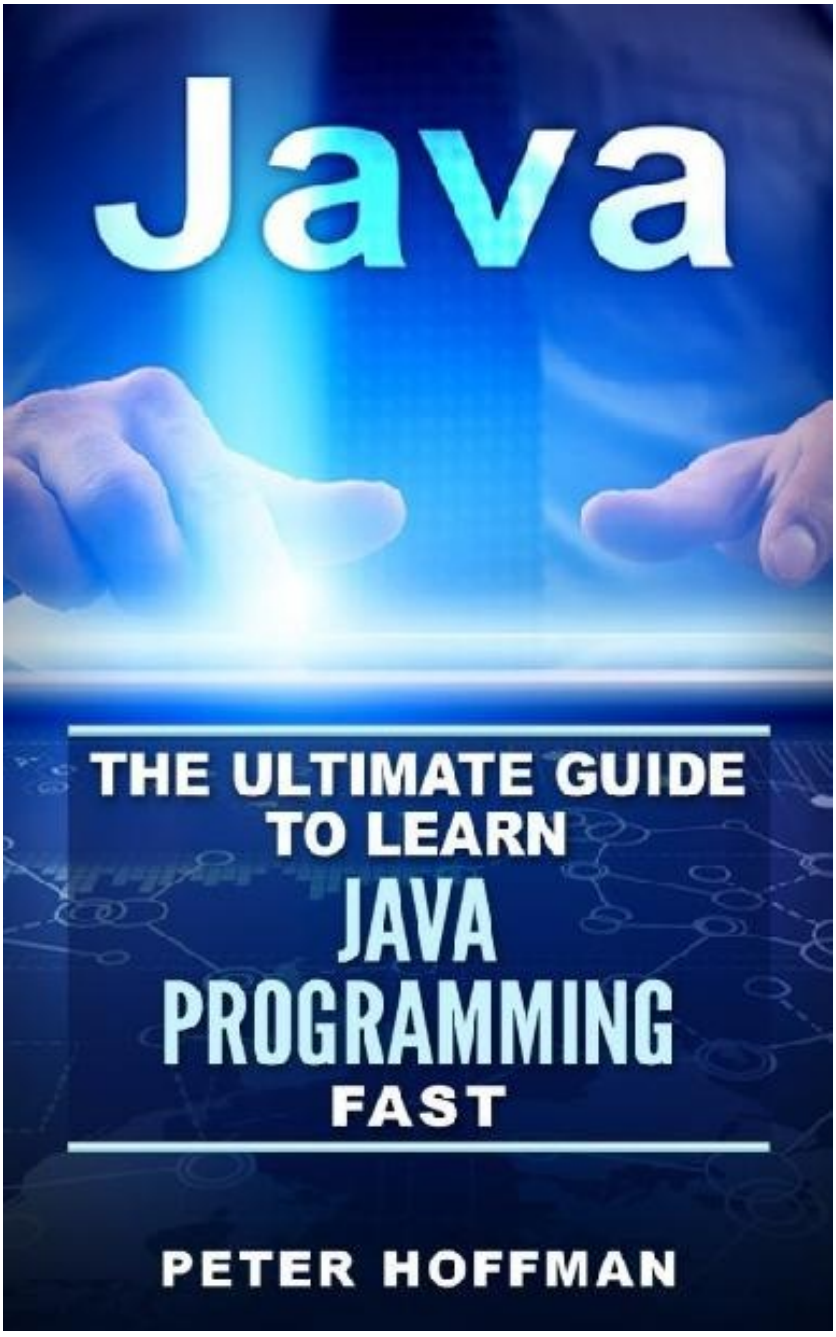

Java
THE ULTIMATE GUIDE
TO LEARN
JAVA
PROGRAMMING
FAST
PETER HOFFMAN


MATT
BENTON
COMPUTER
HACKING

Java

The Ultimate Guide to Learn Java Programming and Computer Hacking (java for beginners, java for dummies, java apps, hacking, how to hack, hacking exposed)


Java
THE ULTIMATE GUIDE
TO LEARN
JAVA
PROGRAMMING
FAST
PETER HOFFMAN

Java

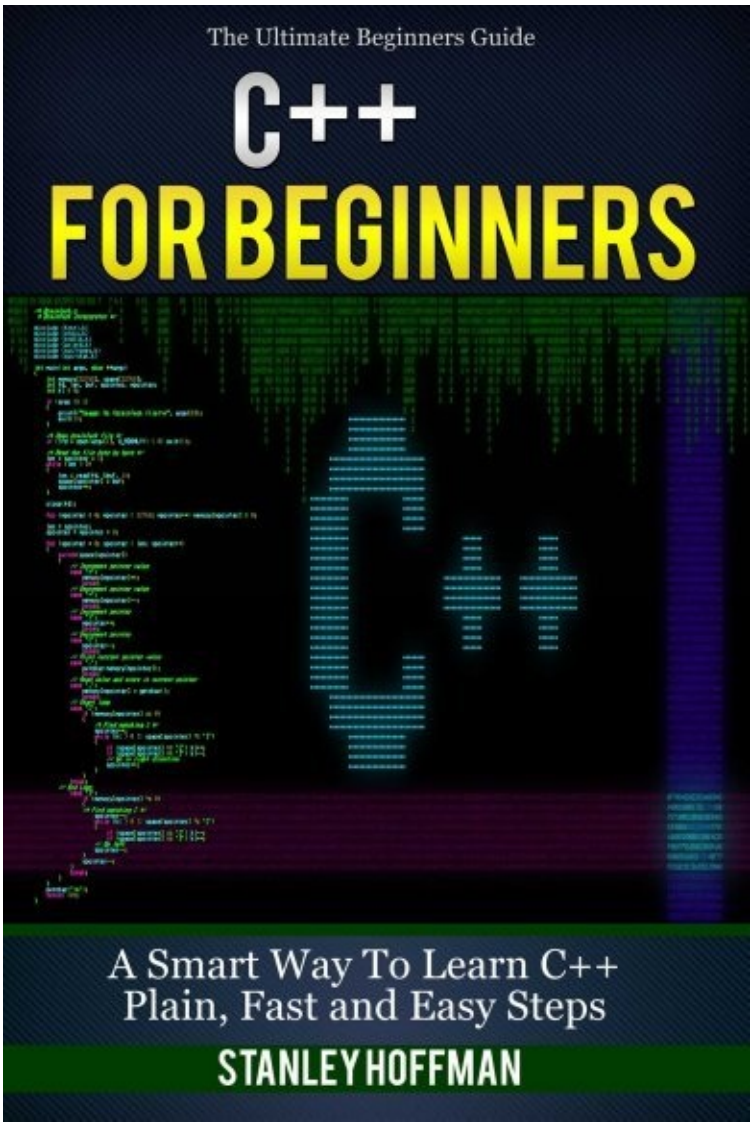
The Ultimate Guide to Learn Java Programming Fast (Java for Beginners, Java for dummies, how to program, java apps, java programming)

PETER HOFFMAN

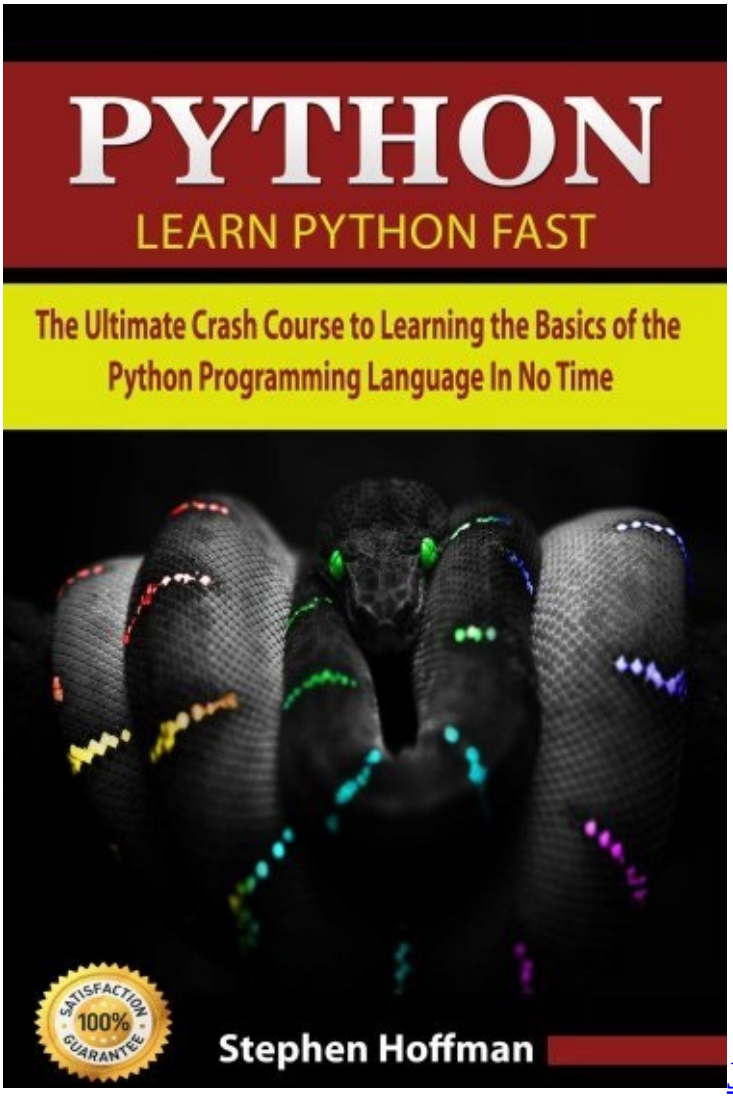
CONTENTS

I think next books will also be interesting for you:

C++

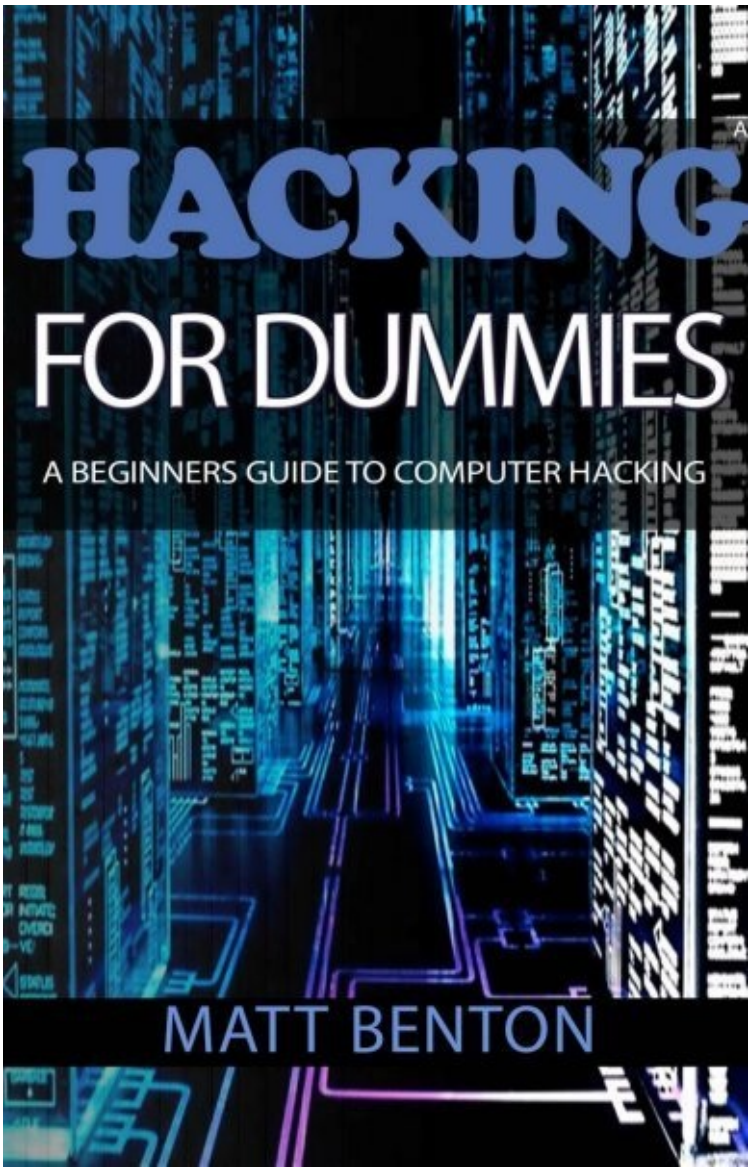


Python

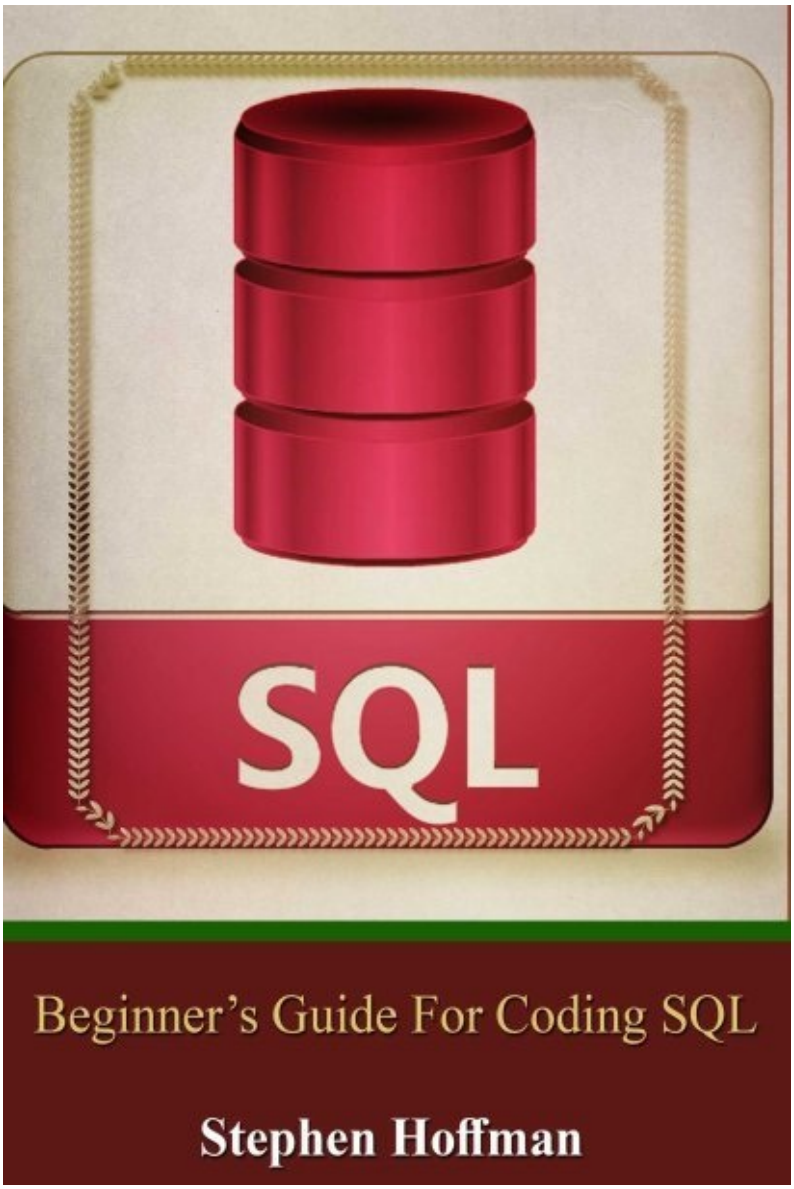


Javascript

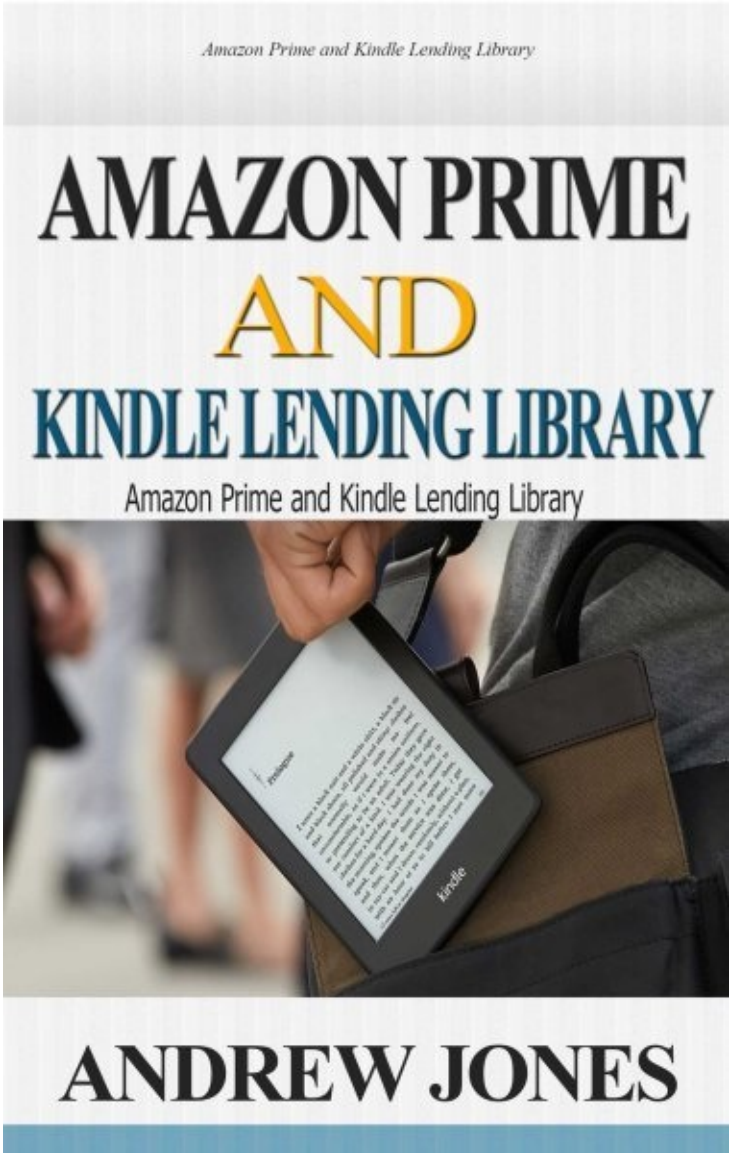
Hacking for Dummies



SQL



[Amazon Prime and Kindle Lending Library]



Introduction

The company behind the Java programming language is the Sun Microsystems. The language was initially introduced in the year 1991. It was developed by a team of people consisting Mike Sheridan, Ed Frank, Chris Warth, Patrick Naughton, and James Gosling. Due to its object-oriented nature, the Java language can be reused with regards to data focus, classes creation, written instructions, and the creation of both instances and objects.

The Four Java Platforms

There are four primary platforms in the Java programming language. They are as follows:

- Java Standard Edition (Java SE)
- Java Enterprise Edition (Java EE)
- Java Micro Edition (Java ME)
- JavaFX

Java-based applications that were being used for computer software were developed on the standard edition of Java. Java-based applications for web servers were developed on the enterprise edition while Java-based applications for multimedia platforms were developed on JavaFX. Most JavaFX-based applications are what we usually call flash players. Lastly, Java-based applications for mobile devices were developed on the micro edition Java platform.

Tools for Beginners

In order to begin your journey to the world of Java programming, you need to choose which IDE to use. IDE, also known as Integrated Development Environment, serves as your workspace in writing Java applications. Imagine it as your notepad in writing HTML codes. However, IDEs were specifically designed for the Java language which means it features all the tools you might need in Java development.

Here are the following IDEs that you can use:

- BlueJ – this is an ideal IDE for beginners. You can easily navigate due to its simplistic interface. It was also originally designed to help beginners in familiarizing the primary concepts of the Java language.

- DrJava – this is a lightweight IDE. Similar to BlueJ, DrJava was also designed for beginners. Although it features a simplistic user interface, DrJava has enough tools that every advanced Java developer can use as well.

- Eclipse – one of the popular IDEs today, the Eclipse is an open source platform with responsive navigation and design. Apparently, it has the cleanest user interface compared to other IDEs in this list.

- NetBeans – this is another popular IDE which is commonly used in Java development nowadays. This is highly recommended for beginners due to its fast and powerful tools that are compatible in all platforms of the Java language.

Each IDE has a unique user interface and ease of navigation. Feel free to use any IDE where you feel most comfortable with. Keep in mind that all of them use the same language although they may appear to be different from one another.

Aside from Java IDEs, another tool you need to begin learning the Java programming language is the JDK. JDK, which stands for the Java Development Kit, serves as your computer's interpreter and compiler for the Java language. Without this, your computer will not be able to translate Java codes into the

universal machine language. All versions of JDK are available from the official website of Oracle.

Take note that you need to choose the JDK version that is most appropriate to your computer’s operating system to avoid future issues when compiling written Java programs.

Misinterpretations

Before we proceed, I would like to clarify that the Java language is not related to the JavaScript language. These are two different programming languages whereas Java is used for the development of stand-alone applications. On the other hand, the JavaScript language is an integrated language for web development. It is used to assert functions that are not achievable through HTML language alone.

This book is about the Java programming language and not about the JavaScript language.

Chapter 1 – Writing a Java Program

Once you are done from choosing your preferred IDE and from downloading the latest version of JDK, you now have the tools to begin programming using the Java language. However, having the tools does not mean that you are already prepared to jump into writing your very first Java application.

It is also important to understand the structure of a Java program – its primary components and their components in the Java application.

Software Types

You can create two types of software using the Java programming language – applets and applications.

Applets are smaller pieces of programming codes that were designed to run on web browsers. We can consider the applets as smaller and lighter applications. They are mainly used to provide either navigation enhancements or additional interactivity to the browser. In contrast with stand-alone Java applications, applets do not need any interpreter in order to execute.

Another type of software that you can develop through Java is a console application. Console applications are stand-alone programs that run within a console environment. All IDEs feature an integrated console environment in order for you to test these kinds of programs. As a beginner, it is important for you to learn console operations to have a good grasp on the basics of Java

language.


<%java
while(wake == true) {
System.out.println
}
if(request.getParameter("age") < 18){
StringBuffer sb = new StringBuffer();
i++;
j--;
sb.append("You can't see the babes!");
out.println(sb.toString());
System.out.println("Bring an extra $500"); break;
System.out.println("Bring an open mind"); break;
aux = i / 10;
for(int aux=1; aux<1000; a
System.out.println("Ple
System.out.println("Hel
}
while(!weekend){
jumpToNextDay();
}
import w
for(long j=1; j<10000000L; j++){
System.out.println("Repeat... I'm not crazy
LETE FROM boss";

Classes, Instances, and Objects

The classes, instances, and objects are the three vital elements of a program created using an object-oriented language such as the Java. In order to understand these three, you need to first look at their relationship from one another.

Classes are the highest and they cover everything in Java programming language. Specific elements in a class are called Objects. Once you add the additional specification to an Object, then it becomes an Instance.

In an analogy, for instance, let us say our class here is the animal. Under animal class, we have different kinds of animals like dogs, cats, birds, and so on. These are the objects under animal class. Under the category of dogs, we have different breeds like poodles, golden retrievers, Labradors, and so on. These are the instances of the object dogs.

Project Creation

The first step in writing your Java program is to start a new empty project in your chosen IDE. You can do this by navigating through the top most toolbar as follows: *File -> New Project*

After selecting New Project, a new window will appear where you will be able to choose from a list of categories. Under the Categories list, select Java and then select Java Application from the adjacent window panel. Afterward, click on the Next button and you will be asked for the project name and the location where you want to save your project.

Always remember the project name since you will be asked to go back to the same project along the way.

Main Class Creation

After the creation of a new project, the IDE will automatically create both the package and the main class. Packages, also known as source packages, serves as folders where you can efficiently organize all assets of the Java project such as classes and instances. Take note, however, that packages are simply identifiers for the organization of assets. Consider them as folders in your computer where you organize your files accordingly.

There should be a main class under the package. If the IDE did not automatically create the main class, then you can create it by navigating on the main menu as follows: *File -> New File -> Java Class* The main class has its vital role in a Java program. The main method, which is what initially executed when opening a Java program, is contained in the main class. In other programming languages, the main method is known as the entry point.

Writing the Package Statement

Now that we already have the main class in our Java project, we are now ready to write our first Java programming language line – the package statement. This statement is an optional part of the program and is only introduced to you for learning purposes. This is written using the following syntax: package [package name]; The IDE will automatically look for the default package if this line is not written. In other words, the package statement might come in handy if you are working on a Java project with multiple packages. Keep in mind that package name must match with the name of the package where the class you need belongs to. Let us say, for instance, that the name of your package is "javatestproject". The syntax should look like this in your IDE:

```
1

2   project javatestproject;

3
```

Writing the Class Declaration

Next to the package statement is the class declaration. The syntax we need to follow here is this: [access modifier] class [class name]

There are two kinds of access modifiers that you can use: public and private. The private modifier prevents the other classes from accessing the declared class while the public modifier allows other classes to access the declared class. For the meantime, let us focus and use the public as our access modifier for our first Java application.

Class name, as what the name suggests, is the name you designated to the main class. Take note that the Java programming language is case sensitive so make sure that you name of the main class with correct upper case and lower case characters. Let us say that the name of our main class is “JavaTestProject”, then this is what should we have right now:

```
1
2   package javatestproject;
3
4   public class JavaTestProject {
5
6   }
7
```

The open ({) and close (}) curly braces refer to the opening and closing of a Java code block. This helps the program in determining where a certain block of

code begins and where it ends.

Writing the Main Method

Now that we are done declaring the class, it is now time to declare the main method for our first program. In our main method, we will be using two keywords: void and static.

The void keyword signals the Java Virtual Machine, also known as JVM, that the program successfully ran and finished. The keyword static signals the program that the field refers to the class and neither the objects nor the instances in it. In other words, the program will be able to freely go through the class without creating instances in that class. Due to the nature of the main method to act as the program's entry point, it is essential to declare it as static. Hence, our syntax for the main method declaration should follow this format: [access modifier] static void

However, this is not a complete declaration for the main method yet. We should also add the parameter "main(String[] args)" to allow the program in executing command-line arguments. And so, our final code for the main method should look like this: public static void main(String[] args) If added to our line of codes above, our IDE should look like this:

```
1

2   package javatestproject;

3

4   public class JavaTestProject {

5

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

7
```

```
8       }
9
10   }
11
```

Writing the Program Output

As of now, you already covered the basic parts of our Java program. Unfortunately, we are not yet telling the program what exactly it needs to do.

What we want for the program to execute is to deliver a message. This is what we call as the Program Output or Output Stream. In order for the program to deliver a message, we should use the following syntax: System.out.print("[message]");

Instead of the parameter "print", you may replace it with "println". This will print the output to a new line once executed. Hence, the alternative format should look like this: System.out.println("[message]");

Let us now tell the program to print the statement "Java is Awesome!" using the command line above. Our syntax should be like this: System.out.print("Java is Awesome!");

And so, our IDE should look like this once we insert the output code:

```
1

2   package javatestproject;

3

4   public class JavaTestProject {

5

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

7
```

```
8           System.out.print(“Java is Awesome!”);
9
10      }
11
12  }
13
```

Allowing User Input

The Java programming language allows you to let users communicate with the program by sending inputs. We can do this by using the built-in class known as the Scanner.

The Scanner class collects the user's input from the computer's keyboard. It stores the collected data into variables that can also be executed in the program's output stream. In order to do this, we need to add the following lines right after the package statement: import java.util.Scanner; This is an example of an import statement which signals the program to import the Scanner class from the built-in Java package known as java.util. This is where we add them in our IDE:

```
1

2  package javatestproject; 3    import java.util.Scanner;

4
```

This, however, does not do anything yet. We need to assign a variable name to the scanner class first in order to call the scanner's collected data into our output stream. We can assign a variable name to the scanner class by using the following statement: Scanner [variable name] = new Scanner(System.in); You can assign any name to the variable. It is a common practice to start the name with an uppercase character followed by lowercase characters. It is also a good practice to separate words by using uppercase characters. Let us say that we want to name our scanner variable as "userinput". Instead of writing the entire variable name in lowercases, replace the first letters of the words "user" and "input" to uppercases. Hence, the statement will look like this: Scanner UserInput = new Scanner(System.in); We should insert the statement within the blocks of code after the declaration of the main method. Hence, our IDE should be as follows:

```
1
```

```
2  package javatestproject; 3    import java.util.Scanner;
4
5   class JavaTestProject {
6
7       public static void main(String[] args) {
8
9           Scanner UserInput = new Scanner(System.in);
10
11      }
12
13  }
14
```

With this set of Java codes, we are commanding the program to assign the user input to the variable named as “UserInput”. Let us now tell the program to call the input. As an example, let us try to ask the user’s name through our program and let the program itself return the user’s name by delivering another output. Our program should be following this schema: First Output -> Input -> Second Output In order to create the first output, we will be using what we have learned from writing an output. Let us ask the user’s name by asserting the following statement above the scanner assigns variable name statement: System.out.print(“What should we call you?”); Next, let us return the user’s answer by calling the collected data into the second output. We can do that by

writing the second output below the scanner variable name statement. Our second output should be: System.out.println(“Hello “+UserInput.nextLine()); With this, we can come up with our final Java program written as follows:

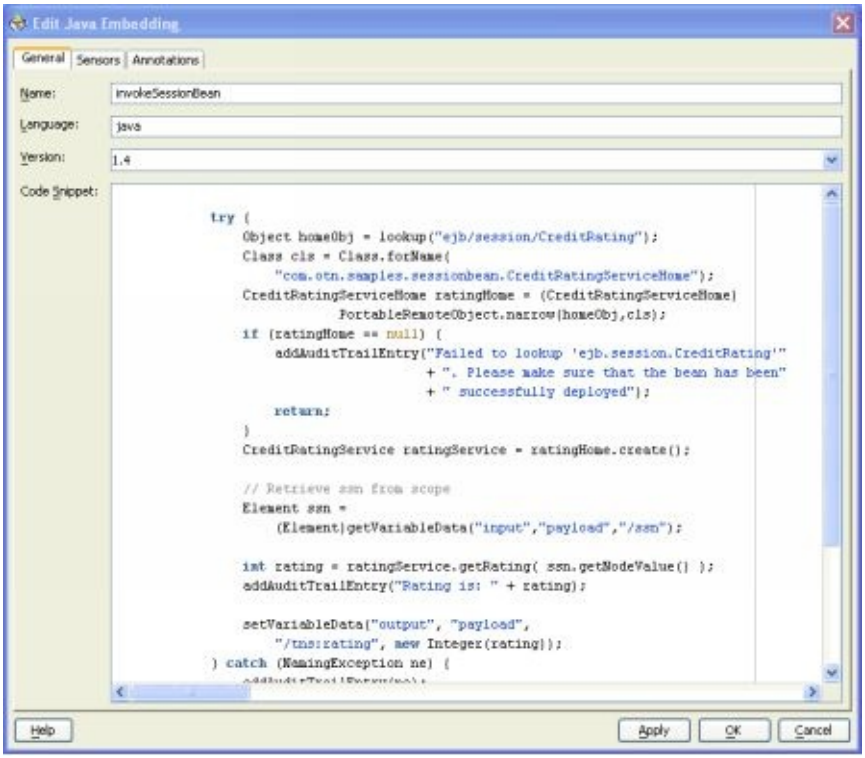
```
1

2   package javatestproject; 3   import java.util.Scanner;

4

5   class JavaTestProject {

6

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

8

9           System.out.print(“What should we call you?”); 10          Scanner UserInput = new Scanner(System.in); 11          System.out.println(“Hello “+UserInput.nextLine());

12

13      }

14

15  }

16
```

Chapter 2 – Variables

Due to the introduction to the scanner class above, we are now familiar with variables. However, the full potential of variables was still unknown to us.

Variables serve as containers of data. They allow us to call specific data in a more convenient way. We can also create a new set of data using inconsistent values through the use of variables. Most programming languages use variables and developers were able to solve both arithmetic and logical operations through them.

In order to utilize a variable, we need to declare it first and then assign a data type upon its declaration. By doing so, the program will be able to recognize what kind of value was stored inside a variable.


Edit Java Embedding
General
Sensors
Annotations
Name:
invokeSessionBean
Language:
java
Version:
1.4
Code Snippet:
try {
Object homeObj = lookup("ejb/session/CreditRating");
Class cls = Class.forName(
"com.otn.samples.sessionbean.CreditRatingServiceHome");
CreditRatingServiceHome ratingHome = (CreditRatingServiceHome)
PortableRemoteObject.narrow(homeObj,cls);
if (ratingHome == null) {
addAuditTrailEntry("Failed to lookup 'ejb.session.CreditRating'"
+ ". Please make sure that the bean has been"
+ " successfully deployed");
return;
}
CreditRatingService ratingService = ratingHome.create();
// Retrieve ssn from scope
Element ssn =
(Element)getVariableData("input","payload","/ssn");
int rating = ratingService.getRating(ssn.getNodeValue());
addAuditTrailEntry("Rating is: " + rating);
setVariableData("output", "payload",
"/tns:rating", new Integer(rating));
} catch (NamingException ne) {
Help
Apply
OK
Cancel

Overview of Data Types

There are different data types and we can use each type in various operations and functions. Before we can learn how to assign a data type to a variable, it is important to first familiarize ourselves with the data types. Here is a table of the data types we can use for variables in the Java programming language and their brief descriptions:

Data Type	

Description

Boolean	This data type has only two possible values: true or false.

Byte	This data type has the tiniest range compared to all number-based data types. Its value by default is 0 and it may store an 8-bit 2s integer.

Character	This data type only stores a single character. By default, the value will always be an empty space.

Double	This data type can store numerical value of 64-bit IEEE 754 and has a value of 0.0d by default. These are numerical values that have decimal points.

Float	This data type stores 32-bit IEEE 754 which is less specific than the Double data type. By default, data is set to 0.0f.

Integer	This data type is the simplest way to handle numerical values. When not specified, it will automatically translate the value into 0. It does not support decimal points which make it inefficient for precise values.

Long	This data type allows you to store numerical value between -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. Its default value is 0.

Short	This data type allows you to store a numerical value between -32,768 to 32,767. By default, it is set to 0.

String	This data stores alphanumeric values. Empty string variables will give null values by default.

Creation of Variables

We need to write a declaration statement in order to create a variable for a certain data type. Any name can be assigned to a variable which we will also use when calling the stored data in the future block of codes.

Let us create a variable using the integer data type and let us name this variable as “MyAge”. Our declaration syntax should be: int MyAge;

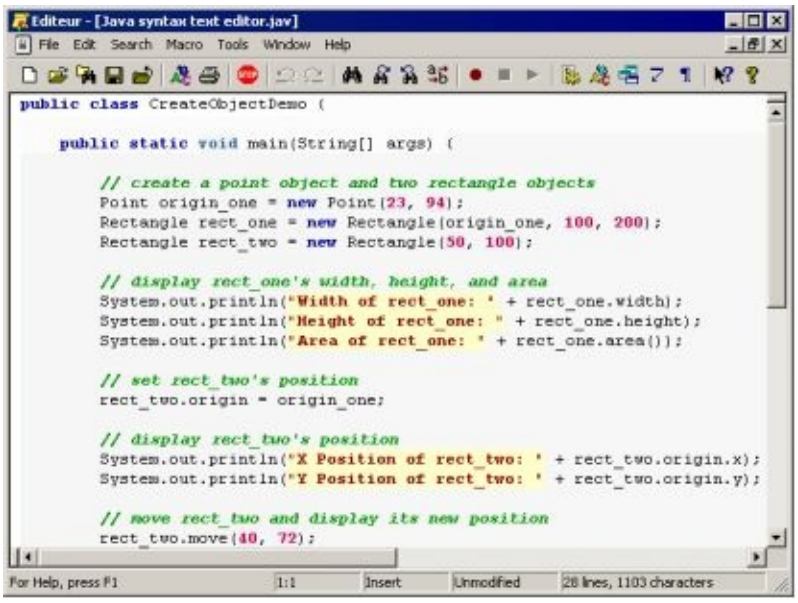
By default, the program will assign the zero value to the MyAge variable since we did not specify an integer value. In order to assign a specific value inside the MyAge variable, we will use the equal (=) operator sign. Let us say that we want to assign the value 20 to the variable. Our syntax will now look like this: int MyAge = 20;

The same goes with other data types. Here is the list of examples for each data type when declaring them in our IDE: (Assigned variable name is “TestVar”)

Syntax Formula	

Example	
Boolean [variable name] = [true or false];	Boolean TestVar = true;
byte [variable name] = [numerical value];	byte TestVar = 0;
char [variable name] = [single character];	char TestVar = L;
double [variable name] = [numerical value]. [numerical value]d;	double TestVar = 31.13d;
float [variable name] = [numerical value]. [numerical value]f;	float TestVar = 1.3f
int [variable name] = [numerical value]	int TestVar = 5
long [variable name] = [numerical value]	long TestVar = 0L
short [variable name] = [numerical value]	short TestVar = 32000
String [variable name] = [alphanumeric value]	String TestVar = null

Data Type and User Input



We already know that we can collect input data with the use of Scanner. Now, we will use the Scanner class to collect other data types. In this example, we will ask for a numerical value from the user and then ask the program to save the value as an integer value.

Our first step is to tell the Scanner class that the value it needs to store is a value of the integer data type. We can do it by changing the nextLine() parameter into nextInt() parameter in our output.

Let us create a program where we will ask the year of birth of the user. Afterward, let us return that value to the user as an integer data type. Our IDE should be like this:

1

2 package javatestproject; 3 import java.util.Scanner;

4

```
5   class JavaTestProject {

6

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

8

9           System.out.print("What year were you born?"); 10          Scanner BirthYear = new Scanner(System.in); 11              System.out.println("You were born in "+BirthYear.nextInt());

12

13      }

14

15  }

16
```

Storing User Input into Another Variable

When calling the Scanner class to collect data from user input, we create a variable as well. In our Java program with regards to the birth year, for instance, we have created a new variable named as “BirthYear” and we store the collected data in it. The problem here is that we also write the parameter right after the variable name when executing the Scanner class data in our program’s output. This is too much hassle when you need to call the collected data in multiple statements. Hence, we can store the entire syntax into another variable to simplify our codes. Do this by declaring a variable while assigning the Scanner syntax as the variable’s value.

Let us create another program where we calculate the user’s age five years from now. This time, we will also store the Scanner call statement into another variable. This is how it should look like:

```
1

2   package javatestproject; 3   import java.util.Scanner;

4

5   class JavaTestProject {

6

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

8

9           System.out.print(“How old are you?”); 10          Scanner UserAge
= new Scanner(System.in); 11          Age = UserAge.nextInt(); 12
System.out.println(“You will be “+(Age+5)+” five years from now”);

13
```

```
14      }
15
16  }
17
```

From our example program, we stored the Scanner call statement into a new variable which we named as “Age”. On the next line, we called the user collected integer and wrote a simple arithmetic operation in order to provide how old our users will be after five years from now. Notice that the mathematical expression is enclosed in parentheses. Without the parentheses, the program will connect the stored value in the Age variable with the numerical value 5 instead of adding the two values together.

In addition to integers, you may also use the following parameters to look for other types of data using the Scanner class call statement:

- nextByte() for Byte data type values
- nextShort() for Short data type values
- nextLong() for Long data type values
- nextFloat() for Float data type values
- nextDouble() for Double data type values

Chapter 3 – Operators

Since the first chapter, we have been using different operators and the most common operator we have used is the assignment operator. It is defined by the equal (=) sign and it signals the program to designate a certain value to a variable.

Aside from the assignment operator, there are five other operators that you use to manipulate data in the Java programming language. They are as follows and their role as mathematical operators:

- Operator for Addition which is represented by the plus sign (+). It commands the program to give you the sum of multiple values.

- Operator for Subtraction which is represented by the hyphen sign (-). It commands the program to give you the difference of multiple values.

- Operator for Multiplication which is represented by the asterisk sign (*). It commands the program to give you the product of multiples values.

- Operator for Division which is represented by the slash sign (/). It commands the program to give you the quotient of multiple values.

- Operator for Remainder which is represented by the percentage sign (%). It commands the program to divide two numerical values and gives you the remainder of the numerical values.

Summary in a Single Program

The best way to familiarize with the operators is by creating a Java program where we can use at least one of the mathematical operators. We will also integrate what we have learned so far. In other words, we will also use the Scanner class to collect data from the user and then execute an arithmetic expression to provide efficient output back to the user.

In this program, we will ask the user to input two numerical values and we will let the program to provide the sum of the numerical values. Let us begin writing our Java program:

```
1

2   package javatestproject; 3   import java.util.Scanner;

4

5   class JavaTestProject {

6

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

8
```

```
9          int Value1, Value2, Total; 10          System.out.print("Please enter the first value"); 11          Scanner FirstValue = next Scanner(System.in); 12          Value1 = FirstValue.nextInt(); 13          System.out.println("Please enter the second value"); 14          Scanner SecondValue = next Scanner(System.in); 15          Value2 = SecondValue.nextInt(); 16          Total = Value1 + Value2; 17          System.out.println("The total of the two values is "+Total);

18

19      }

20

21    }

22
```

In this program, you will notice that we declared three variables as containers for integer values. This was written on line 3 where we used the following syntax: int Value1, Value2, Total; Because of this, we no longer need to enclose the two variables inside the parentheses as we call them for a mathematical expression on line 16.

We also allow users to enter two numerical values by using two Scanner call statements. Then, we stored these values into two separate variables that added to one another inside another variable. This way, we were able to call the sum of the two values for our final output stream.

Conclusion

The Java programming language gives you unlimited tools to achieve different operations. However, it is extremely important to get to know the IDE you chose to use. Beginners should find which IDE they feel most comfortable working on. Although all IDEs use a single programming language, each has its own unique interface that might change your programming pace.

We also learned how to start a new project and the essential elements a project needs such as the main class and the main method. Packages were also introduced and although the package declaration is optional, we have learned that declaring packages in our Java program will help us efficiently organize multiple classes for bigger projects.

The main class and the main method may also contain access modifiers that will allow the program in defining whether it should allow other classes in accessing the statement or not. It is a general practice to use “Public” as the main class’s access modifier.

Allowing the program to return a message as its output has also been discussed in the first chapter of this book. You may use the keyword “print” as the simplest form of the parameter or you may rather sign the Java program to show the string in another line. This can be done with the use of the keyword “println”.

We also discussed how dull it is to have a one-way-only program where users were not capable of sending an input. The Java programming language has an answer to this by allowing you to use its built-in Scanner class which can be

found in the built-in local package known as java.util. The Scanner class allows us to collect data entered by the user from their keyboard. This opened new operations such as returning the user input as the program output.

The next chapter introduces variables and how do they bring out the full potential of the Java programming language. We discussed the different data types that we can store inside variables and how to call these data for our program's output. It is also important to remember that we can also store variables into another variable for efficient programming later on.

Lastly, our last chapter discusses the different operators aside from the assignment operator represented by the equal sign (=). In the end, we combined everything we have learned to create a Java program which can solve a mathematical equation based on user's input.

At this point, I am confident that you already grasped the idea behind the Java programming language as an object-oriented language. I personally congratulate you for finishing this book and I am quite sure that you are now ready to move forward to a more advanced method in Java programming.

Do not forget that you have only took a glimpse on what the Java language is capable of. This book has a continuation where you will learn more advanced techniques such as flow control, a deeper understanding of access modifiers, a closer look to objects and classes, constructors, serials, and inheritance.

Consider this book as your big leap into the world of Java programming language. Always remember that this programming language is fun to learn, yet filled with tons of challenges ahead. Nevertheless, mastering the language will open new opportunities to you such as being able to develop your own mobile application. You may even end up developing a new stand-alone game that may stand the test of time like Minecraft.

MATT
BENTON

COMPUTER HACKING

THE ESSENTIAL HACKING GUIDE FOR BEGINNERS

Computer Hacking

The Essential Hacking Guide for Beginners

MATT BENTON

Contents

Introduction – What is Hacking?

Hacking is the act of gaining unauthorized access to a computer system, and can include viewing or copying data, or even creating new data. Often hacking is

understood to be a way of maliciously disrupting a computer system, copying information, or leaving behind a virus that destroys data.

There are many different reasons why hacking takes place, and these reasons range from wanting to disrupt a system due to ideology (so hacking as a means of protesting); wanting to gain profit for example in order to commit credit card fraud; or simply hacking for the sake of enjoyment and amusement.

There is some controversy about the definition of the word ‘hacker’ because those that try to prevent such breaches in security from taking place, or seek to recover lost files, can also be known as hackers. Thus, some people believe that the correct term for malicious system security breaches is in fact ‘cracking’ and that ‘hacking’ is the correct word to use for those who fight against such malicious exploitation of computer weaknesses.

However, in the popular imagination and in general conversation, the word ‘hacker’ is mainly understood to refer to the ‘bad’ method of breaking through computer security. The two processes share many common skills, as regardless of motivation (whether to steal or protect, break in to or save, computer data) the same understanding of computers is required.

Hacking is more than simply a pastime for those who are interested in technology, and more than simply an illegal activity used for personal gain and with malicious intent, although both of these motivations do make up much of hacking activity. In fact, hacking is its own subculture, and members of the community feel very strongly about their ideologies, techniques and social relationships in the computer underworld.

There are many hacking groups and conventions, such as SummerCon, DEF CON, HoHoCon, ShmooCon, BlackHat, Chaos Communication and Hacker Halted, and local hacking communities take their entries into hacking competitions very seriously. Unsurprisingly there are also numerous online groups and forums dedicated to the subject of hacking, and there is certainly a strong community spirit felt by those with similar hacking ideologies.

Furthermore, hackers are often passionate about literary depictions of the hacking community, and ardently read fictional Cyberpunk and factual hacker

magazines.

This book will serve as an introduction to the world of hacking, and will provide insight into some of the key influences, ideologies, groups, concepts, and techniques of hacking.

The first chapter will consider the beginnings of hacking and the influence of the literary genre, Cyberpunk. The second chapter will look at the different types of hackers, and draw a distinction between ethical and unethical hacking. The third chapter will look at the issue of computer security, which is vital to an understanding of hacking.

The final chapter will provide an overview of the various different techniques for hacking, including automated and manual approaches as well as the importance of the cyber confidence trick known as social engineering.

Chapter 1 – Hacking and the Influence of Cyberpunk

Michael Bruce Sterling, the American science fiction author, helped establish the popular genre of Cyberpunk. Cyberpunk is a subcategory of science fiction that focuses on the role of technology in a future setting. In this literary and cinematic genre, lower-class citizens are depicted, who have access to, and a great understanding of, advanced technology.

Cyberpunk often explores the role of technology during the breakdown of social order, in which there is an oppressive government restricting and damaging the lives of the general population. Furthermore, artificial intelligence (such as robots or intelligent computers) also plays a significant part in Cyberpunk stories, and the Earth is depicted in the near future in a post-industrial dystopia (the opposite of utopia, and therefore a bleak world characterized by oppression and often social unrest.)

The impact of Cyberpunk in the present-day understanding of hacking is considerable. Science fiction is particularly effective when we can recognize our own world within the fictional representation, and with Cyberpunk we can

recognize many of the concerns of the contemporary technological age. Lawrence Person (editor of the science fiction magazine *Nova Express*) describes the typical characters in Cyberpunk: “Classic cyberpunk characters were marginalized, alienated loners who lived on the edge of society in generally dystropic futures where daily life was impacted by rapid technological change, an ubiquitous data sphere of computerized information, and invasive modification of the human body.”

To a contemporary reader, this description of Cyberpunk characters is reminiscent of how hackers are thought of in the popular imagination, and depicted in books and in films. Therefore, the interplay between Cyberpunk characters and how we view real-life hackers is considerable: in many ways our understanding of what a hacker is like is based on how Cyberpunk characters are depicted in fiction. One example of this is how in Cyberpunk the characters often live in filthy conditions, work at night and sleep all day, and do not have any social life beyond chat rooms.

In the present-day imagination when we think of hackers we will often think of a lonely adolescent boy sitting in a darkened room behind a computer screen. In fact, Michael Bruce Sterling, who was one of the first science-fiction writers who dealt with Cyberpunk, has also shown the most interest in understanding the development of hacking.

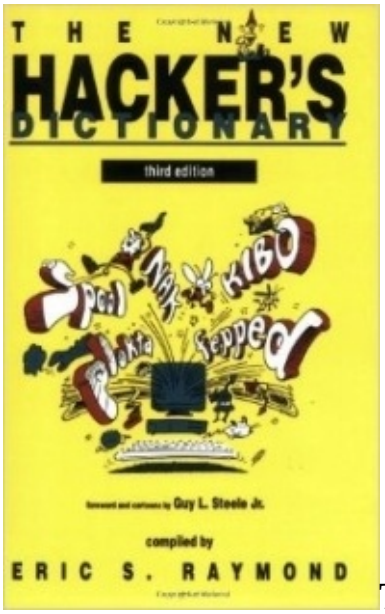
Sterling has traced the emergence of hacking, and the associated underground computer network, to the Yippies, a counterculture group who were active in the

1960s and published *Technological Assistance Program*, a newsletter that taught its readership techniques for unauthorized access to telephones, known as phreaking.

Many of the individuals who were involved in the phreaking community are also an active part of the underground hacking community, suggesting that the relationship between the two groups.

Chapter 2 – The Different Types of Hackers

The computer hacking underground contains various different subcategories of hackers. This is mainly due to conflicting ideologies, whereby certain groups calls themselves by a specific name, or call others a specific name, in order to emphasize that they do not agree with the ideologies of others.



The generic word ‘hacker’ therefore, although referring to those who have technical knowledge and are able to gain unauthorized access to computer systems, is rather vague and does not distinguish between those who use different methods or believe certain things.

Instead, separate names have emerged in order to distinguish between groups,

and to indicate that not all hackers follow the same rules or ideologies. One way in which this can be seen, as discussed previously, is the distinction between hackers and crackers, as advocated by Eric S. Raymond in *The New Hacker's Dictionary*.

In this book Raymond compiled a glossary of hackers' computer programming jargon, but those from the hacking community feel that this book is too biased by Raymond's own view of hacking as a malicious practice.

Rather than following the dichotomy of hacker/cracker that Raymond suggested, the general hacking community feels that this is too reductive and instead advocate a wider list of name to reflect the spectrum of beliefs and practices of the large hacking community.

One subcategory of hackers is known as 'white hat hackers' and they break through computer security without a malicious motivation. Examples of why this might be done include doing so to test one's own security effectiveness, or when doing work developing computer security software.

These breaches of security can occur whilst performing vulnerability assessments of computer software as part of a contractual agreement, and is therefore legal. In this way, the slang term 'white hat' references an ethical hacker who does so for positive reasons, in order to protect rather than destroy. There are recognized organizations, such as The International Council of Electronic Commerce Consultants, who provide training and certificates for this area of ethical hacking.

 On the other hand, there are 'black hat hackers' who breach computer security systems simply to be malicious, or to gain profit.

These hackers are the ones who are also sometimes referred to as crackers. This subcategory form the cliché hackers who are often depicted in films and television, and represent the elusive and little-understood computer criminal who the public fears.

These types of hackers violate computer security in order to destroy, change or steal information, or to prevent authorized users from being able to access the system. In this way they can cause disruption, waste time, and cause distress, but they can also steal significant amounts of money or access confidential information.

Generally a black hat hacker will spend time looking for and discovering faults in programs, or weaknesses in computer systems, but rather than alert the public to these problems they exploit them for personal gain or simply for fun. Once they have accessed a computer system, they can consequently make adjustments that prevent somebody with authorized access from using the system and thus the black hat hackers retain control.

Lying somewhere between the two, not quite a white hat hacker and not quite a black hat hacker, is the gray hat hacker. This is somebody who without being asked to searches the Internet for systems with a weakness or security flaw, and will then notify the administrator and offer to rectify the problem for a fee.

In this way they are not as good as a white hat hacker (because they are demanding a fee, and their services were never requested) but they are also not as bad as a black hat hacker because they do not exploit these weaknesses in order to wreak disruption of steal data. Another way in which gray hat hackers might respond to their discovery of a security weakness is to publish their findings online, so that the general public has access to the information.

In this way they are not performing malicious hacking themselves, but they are publishing the information, which leaves their subject at risk of a security breach. This type of hacking is illegal and also considered unethical, whether or not the gray hat hacker has breached security for personal gain, because they have gained unauthorized access to data and have left the system susceptible to hacking by malicious blat hat hacker groups.

As well as these three main classifications for hacking, which differentiate hackers based on their motivation and what they do about the information they discover, there are various other specific types of hacker. There is a social hierarchy amongst hackers, who are recognized based on their skill.

The highest of these statuses is the elite hacker, and sometimes form into elite groups such as the ‘Masters of Deception.’ On the other end of the scale is a script kiddie, who is still learning and has not yet developed their skills with breaching security systems. A script kiddie uses automated tool written by others, and is therefore simply following a code provided by a more skilled, black hat hacker, and not having to work it out themselves. Usually a script kiddie does not really have any knowledge or understanding of the complicated underlying technological concepts, and simply follows a plan provided by a more experienced hacker.

Even less experience than a script kiddie is a neophyte, who is a completely new hacker who has very little knowledge of computer technologies or the logic and concepts behind hacking. A blue hat refers to somebody who is used by computer security consulting firms but is not actually a part of the company; the blue hat is used to test a system prior to its launch to determine whether it has sufficient security or will be susceptible to hacking.

A hacktivist (a combination of the words ‘hacker’ and ‘activist’) is a hacker who uses their knowledge of technology and their hacking skills in order to broadcast a political, social or religious message. Hacktivism itself has two subcategories: cyber terrorism (where websites are damaged or services cannot be accessed) and freedom of information (making information available to the public that was previously either undisclosed or stored in an encrypted format.) Groups of

hackers working collectively can include organized criminal gangs, and cyber warfare of nation states. The different subcategories of hackers are indicative of the various ideologies, motivations and techniques that are present in the hacking community.

Chapter 3 – Computer Security

Before we can begin to explore the key concepts and techniques of hacking, it is helpful to first understand the basics of computer security. As hacking is the act of breaking through security measures of computer systems, an understanding of these systems is vital to any hacker who hopes to penetrate them. Computer security is applied to computers, smartphones, computer networks (public and private) and the entire Internet in order to protect devices, data and services.

Digital equipment is protected from unauthorized access by computer security, to ensure that data is not stolen, changed or deleted and to maintain the smooth running of systems. In present-day society, where digital culture forever growing, protecting these systems is extremely important and thus the field of computer security is forever growing and developing. Part of computer security is protecting the physical equipment from theft, whereas the other part of computer security is information security, to protect the data itself (and this is where hacking comes into play.) However, sometimes these two fields overlap because if there is a breach in physical security (e.g. if a laptop is stolen) then it becomes much easier for the individual to succeed in a breach of information security, since they have the piece of equipment and it is therefore easier to access data than it is remotely.

Cyber security encompasses all security measures in place to protect a computer's data, and includes procedures such as awareness training, penetration testing, and the use of passwords to confirm authorization in order to protect data both when it is in transit and when it is simply being stored. The financial cost of being a victim of a computer security breach is considerable and as a consequence there is a lucrative market for anti-virus and computer security protection.

Computer security is a huge field because of our present-day reliance on technology. Almost every industry uses computers to a greater or lesser extent, and therefore the extent and variety of computer security measures is vast. There are some areas, however, where computer security is particularly important because they are especially vulnerable to breaches in security.

One of these is the area of financial systems, because hackers can make a profit by stealing data and consequently accessing funds. Any website that requires somebody to enter their credit card numbers are often targeted because a hacker can immediately transfer money to their own account, or spend the victim's money online.

Even if the hacker themselves do not directly use the person's bank details, they may also sell the information illegally, in order to distance themselves from the crime and attempt to avoid being caught. It is not only online that a person's data can be stolen; in-store card machines and cash points can also be rigged to collect personal information and thus gain access to funds.

People are becoming increasingly aware of this risk when doing online shopping or entering their card details, and therefore various measures are being put in place including using passwords and answering security questions. The aviation

industry is another field in which computer security is of the upmost importance, because the consequences of a breach in security can range from the publication of confidential information, to the loss of expensive equipment and human life.

There are various reasons why the aviation industry may become subject to a computer security attack, depending on the motivation for the crime. These motivations include sabotage and espionage in the military aviation industry, and industrial competition and terrorism in the commercial aviation industry. Air traffic control is one of the aviation industries most vulnerable systems, because any attack can be difficult to trace, and are relatively simple because it only requires a spoof message on the radio.

There are those who seek to exploit computer vulnerabilities (either due to thrill seeking, to make a political/social statement or for financial gain) and of course on the other side those who work to uphold computer security again such threats. Somebody with knowledge of technology, and the ability to hack into computer systems, can therefore either become involved in the illegal and unethical form of hacking (otherwise known as cracking), or serve the other side by identifying threats, improving security measures and alerting companies to the vulnerabilities in their systems.

For those whose aim is to protect computer security, there are various countermeasures to guard against damaging hacking, whereby the risk of being vulnerable to a breach of computer security can be minimized or eliminated. These precautions vary in cost and complexity, but can include: intrusion detection systems (to detect threats and also analyze attacks after the event), the use of account controls (passwords and encryption of data); and the installation of firewalls (providing either hardware or software package filtering of certain forms of attack.

Precautions against a computer system being compromised by attack include making steps to prevent attack, ensure any potential attacks are detected, and the ability to respond to an attack to prevent further damage.

However, despite there being a range of countermeasures available, computer systems still remain vulnerable to attack and it is certainly not uncommon for a computer to have its security compromised. The first reason why attempted security violations still occur is that the police are often unfamiliar with computer technology and as a result do not have either the skill or the inclination to solve the crimes and apprehend the criminals responsible.

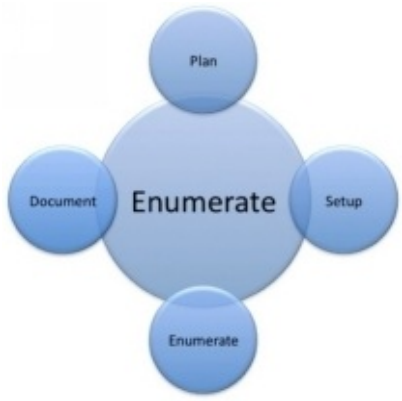
Furthermore, any investigation of such matters requires a search warrant in order for an officer to examine the entire network and this can make the procedure extremely time consuming. Another difficulty in ensuring computer security is that in the age of globalization, in which information can be shared throughout the world using the internet, and technology can spread data extremely easily, identifying and apprehending those responsible is particularly difficult.

The reason for this difficulty is that a hacker might be working from one jurisdiction, while the system they are hacking into is in a different jurisdiction. Furthermore, a hacker can use various techniques (such as a temporary dial-up internet) in order to ensure their anonymity. The third problem is due to the high number of attacks that occur.

Organizations can be subject to many attacks and therefore are unable to pursue every security threat. A computer user would benefit from taking precautionary measures in order to ensure their computer security, as once a breach of security has occurred there is not much that can be done to rectify it.

Chapter 4 – Hacking Techniques

There are various techniques that can be used by hackers in order to gain unauthorized access to a computer system, in order to wreak havoc, steal money or data, or to prevent the system from operating as it is supposed to. The three main methods that are used in order to attack a system that is connected to the Internet are: network enumeration, vulnerability analysis and exploitation.

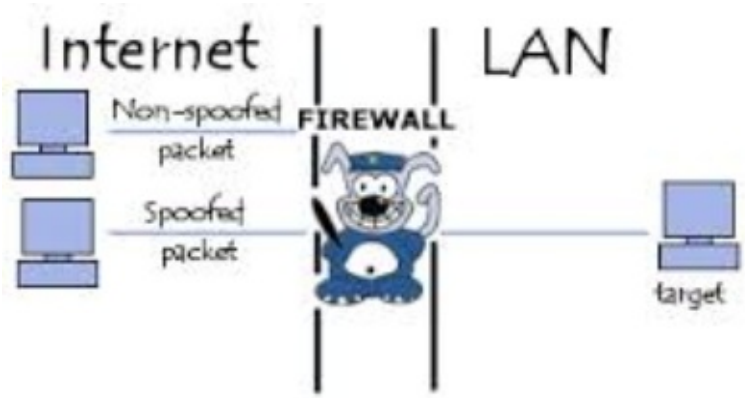


A network enumerator is a program that is used in order to discover the usernames and other information from networked computers. The program discovers any weaknesses in the computer network's security and the findings are reported to a hacker who may then use this information in order to access the network and cause damage (either by stealing data or corrupting the network.) On the other hand, ethical hackers can use the same process simply to discover any weaknesses in their system in order to tighten security. Another method used is vulnerability analysis, which identifies any points of vulnerability in a system; this information can then be used to

either attack the system, or to remove the weakness. Vulnerability analysis can then lead to exploitation, where the hacker uses the vulnerability information in order to breach a computer or system's security.

There are many specific techniques that can be used, but they all employ the main concepts and methods described above. The first more specific example of a hacking technique is a vulnerability scanner, which is a program used to check a network for susceptibility to attack. A port scanner can also be used, which identifies avenues of access to a computer and can establish how to circumnavigate a firewall.

As well as these mechanized devices, hackers can also find these vulnerabilities themselves, which can be done by manually searching the code of the computer and then testing whether they are right. Brute-force attack is another method by which a hacker can gain unauthorized entry to a computer network, and this involves for example guessing passwords. Password cracking is another hacking technique that uses passwords, but rather than guessing the password, the hacker recovers password information that has been stored in the computer, or transmitted.



A spoofing attack (otherwise known as phishing) is an enemy program, system or website that poses as a trusted one. By falsifying data the hacker is able to masquerade as a trusted system and thus fool a program or user into revealing confidential information such as passwords or bank details. Another hacking technique that is commonly used is a root kit, which is a program that manages to take over the control of an operating system by employing hard to detect methods.

A Trojan horse is yet another technique that is a program which manages to fool systems and users; it works by working in one way while seeming to be doing something else. By using this method a hacker is able to gain unauthorized access to a system and create an access point so that they can re-enter via that established route later on. A computer virus is the most widely recognized form of hacking, as it is the computer threat that most of the public is aware of.

The virus works by self-replicating and implanting itself into documents and code; while some computer viruses are malicious some are merely irritating or harmless. A computer worm is similar in that it is self-replicating, but it is able to enter a computer program without a user inadvertently letting it in, and it does not need to insert itself into present programs.

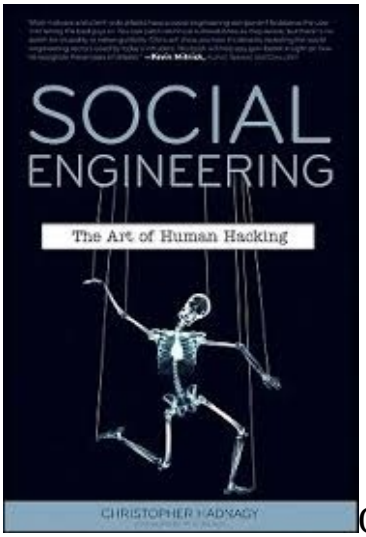
Finally, a keylogger is a tool that records every keystroke on a given machine, which can later be accessed and viewed by the hacker. This is usually to enable the hacker to access confidential information that has been typed by the victim. In fact, there are some legitimate uses for such a technique, for example some companies use a keylogger in order to detect any dishonesty or fraud committed by an employee.

A large area of computer hacking involves the use of social engineering, whereby in order to circumvent information security a person is manipulated in order to reveal confidential information or to grant access to secure networks. This technique (which includes phishing) is usually only part of a complex routine in a wider fraud scheme, but it is also a dangerous step because human beings are more likely to be won over by a convincing trickster than a machine is.

Social engineering relies on the psychological act of decision-making, and can be thought of as one of the most significant vulnerabilities in a computer security system. There are many different ways in which social engineering can be applied in order to gain unauthorized access to a computer system, and this

includes criminals posing as IT technicians who pretend that they are fixing the company computers whilst in fact stealing data.

Another example would be a trickster informing a company that the number of the IT helpdesk has changed, so that when employees phone the number they will willingly disclose their account details thinking that they are talking to somebody who they can trust with the information. These sorts of scenarios come under the category of ‘pretexting’ because making up a believable scenario allows the criminal to access the required information and this leads the victim to disclose the information.



Other professionals that a hacker involved in social engineering could pose as include the police or bank manager, because these are individuals who we believe have the right to be granted any information that they request. Baiting is a subcategory of social engineering because it relies on human psychology in order to work. Baiting is where a victim’s computer security is compromised when an infected disk, device or USB stick is used.

An example of baiting would be for the criminal to post a USB through somebody’s door with a tempting sounding label and simply wait for the curious victim to plug it into their laptop, at which point malware would automatically install and infect their computer. This technique makes the most of the human tendency towards curiosity and greed, because if a label promises erotic images, money or gossip then a victim may find it hard to resist taking a look.

Kevin Mitnick, a once computer criminal who later because a security

consultant, has pointed out that it is much easier and quicker to trick a person into disclosing confidential information than it is to crack into the system using luck, brute force or technical knowledge. Christopher Hadnagy has written a book titled *Social Engineering: The Art of Human Hacking, which* emphasizes the way in which humans are the most vulnerable part of any computer system.

Conclusion

This book has provided an overview of some of the key concepts to do with hacking. We have considered the beginnings of hacking and how it was influenced by the literary tradition of Cyberpunk. It is interesting to note that what was once depicted in science-fiction as an imagined activity in a dystopian society has become real and has gone on to pose a significant threat to computer security and a central concern of information technology experts.

Here we can see the true beginnings of hacking and how what was once a fictional theoretical concept has become a reality with a significant impact on the digital culture. Next we looked at the different types of hackers and noted the distinction between ethical hackers, who perform hacking legally and in order to improve computer security (otherwise known as white hat hackers) and unethical hackers, who use their skills illegally in order to wreak havoc, disrupt services, and steal information and data (otherwise known as black hat hackers.) With this it is interesting to note how technical skill can be used differently depending on motivation, and how the hacking community is not united by a clear and consistent ideology. Subsequently we looked at computer security in order to better understand the conditions within which hacking takes place. By learning about computer security it is possible to understand the challenges faced by hackers who come up against security measures, as well as the challenges faced by those seeking to maintain the security of their computer systems.

Any introduction to hacking would not be complete without this examination of computer security because the value of maintaining computer security is what motivates ethical hackers, and what unethical hackers are fighting against. Finally we looked at the numerous different hacking techniques that can be utilized, including both automated software that finds and exploits vulnerabilities in computer systems, as well as manual methods for breaking through security measures such as discovering a password through trial and error.

In this chapter we also considered the vast area of hacking that is social engineering, by which a hacker is able to access a secure network by illegally obtaining the information needed. In today's digital culture we are becoming increasingly reliant on technology for everything that we do.

As technology has improved our lives have become easier in many ways, but with this blossoming industry new types of criminals have also been created. A criminal does now not even need to leave the house in order to steal money, but can do so simply by hacking into their victim's computer and accessing confidential data.

Hacking is not always illegal though, and this book has also looked at the ways in which there is an increasing demand for computer experts to become ethical hackers in order to further promote and protect computer security. This introduction to the world of hacking has revealed that hacking is not a simple activity but a huge spectrum of different behaviors that involve a wide range of techniques and motivations.

Moreover, hacking is shown to be an activity that has a strong sense of cult affiliation, in the sense that hackers strongly feel part of the hacking community. As digital culture continues to grow, it seems that both ethical and unethical hacking will become more and more skilled and its impact evermore significant.

Thank you for reading. I hope you enjoy it. I ask you to leave your honest feedback.