

Introduction to Java

Chapter 1 Homework



September 4, 2019

Thomas J. Short

Cis 260-1

**Chapter 1 Exercises**

**Intro to Java Programming 11e**

1.2.3. The Acronym CPU is short for **Central Processing Unit**. The Computer has an internal clock that uses the measurement **hertz (HZ),** with one HZ equaling 1 pulse per second. Today we use Gigahertz, whereas, older CPU’s used the measurement Megahertz.

1.2.4. bit - A bit stores a value in the form of a binary digit in the binary number system represented by either the number (0) for off or (1) for on.

Byte- A byte is the minimum storage unit in a computer, comprised of a series of 8 bits. Series of bytes are encoded and decoded thru an encoding scheme

1.3.1 The Central Processing Unit (C.P.U) only understands **Machine Language** which is written in binary code, a series of ones and zeros.

1.3.3 high level Programming language are platform independent, English-like and are easier to learn then assembly language or Machine code. Programs written in a high-level language’s like Java or Python are called **source code** or a source program.

1.5.3 Android use’s **Java** to write source code.

1.6.3 IDE stands for **Integrated Development Environment**, such as, NetBeans, Eclipse and BlueJ.

1.7.2 Java is **case sensitive** and Reserved words(keywords) are **lower case**, but by convention class names start with capital letters.

1.8.1 The Java source filename extension is **.java** and must have the exact name as the public class name. Once the Source code is compiled the compiler generates a bytecode file with a **.class** extension.

1.8.2 A **Java source code file**, such as, Welcome.java is **inputted** into the compiler and it generates(**outputs**) a **Java bytecode executable file**, such as, (Welcome.class). The byte code file can now run on a Java Virtual Machine (JVM) on any computer

1.8.6 Java is **platform independent** meaning it can run on **any machine**. To run Java on a computer you need a **Java Virtual Machine** (JVM)

1.10.1 Syntax errors are errors in code construction, such as, selling of keyword, punctuation, and the opening and closing of braces. Runtime errors happen when the user enters a value the program can’t handle. For example, if the program expects a number and receives a string or division by zero. This usually results in the program crashing.

1.10.5 This is an example of a **logic error** because the program did not do what it was intended to do. These errors are often the most difficult to debug.

1.10.6 public class Welcome {

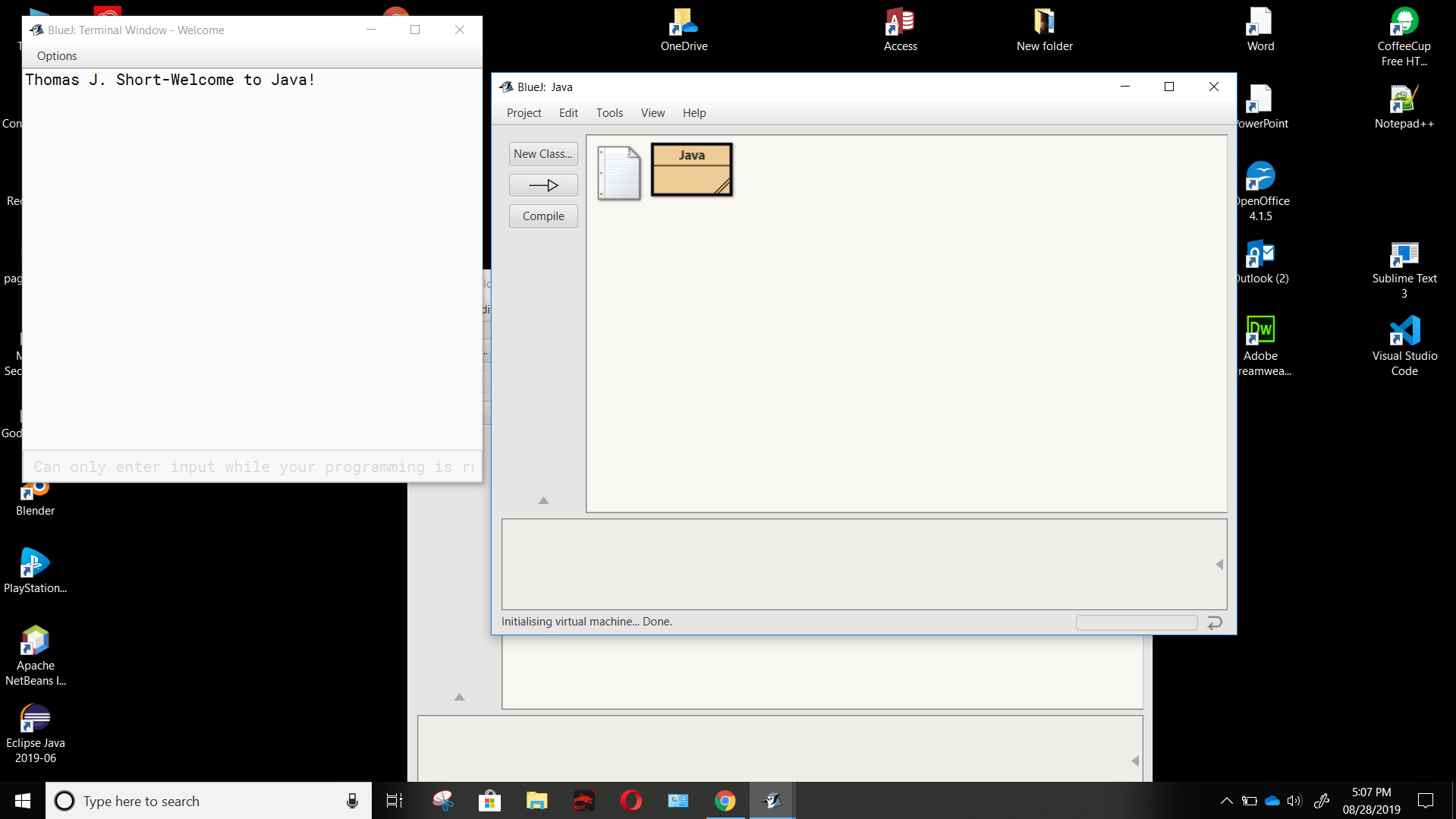
Public void Main(String[] args) { 1. identifier expected (static) 2. main is capitalized

System.out.println(‘Welcome to Java!); 3. Illegal character ‘ , should be (“Welcome to Java!”);

}

) 4. Reached end of file while parsing, should be }

**Example 1.7**

**Listing 1.1**

**Programming Exercise 1.3**

