

CHAPTER 01 | Java exercise from Deitel

1.1 Fill in the blanks in each of the following statements:

- a) Computers process data under the control of sets of instructions called **programs**.
- b) The key logical units of the computer are the **input unit**, **output unit**, **memory unit**, **central processing unit**, **arithmetic and logic unit** and **secondary storage unit**.
- c) The three types of languages discussed in the chapter are **machine languages**, **assembly languages** and **high-level languages**.
- d) The programs that translate high-level language programs into machine language are called **compilers**.
- e) **Android** is an operating system for mobile devices based on the Linux kernel and Java.
- f) **Release candidate** software is generally feature complete, (supposedly) bug free and ready for use by the community.
- g) The Wii Remote, as well as many smartphones, use **accelerometer** which allows the device to respond to motion.

1.2 Fill in the blanks in each of the following sentences about the Java environment:

- a) The **java** command from the JDK executes a Java application.
- b) The **javac** command from the JDK compiles a Java program.
- c) A Java source code file must end with the **.java** file extension.
- d) When a Java program is compiled, the file produced by the compiler ends with the **.class** file extension.
- e) The file produced by the Java compiler contains **bytecodes** that are executed by the Java Virtual Machine.

1.3 in the blanks in each of the following statements (based on Section 1.5):

- a) Objects enable the design practice of **information hiding** although they may know how to communicate with one another across well-defined interfaces, they normally are not allowed to know how other objects are implemented.
- b) Java programmers concentrate on creating **classes**, which contain fields and the set of methods that manipulate those fields and provide services to clients.
- c) The process of analyzing and designing a system from an object-oriented point of view is called **object oriented analysis and design (OOAD)**.
- d) A new class of objects can be created conveniently by **inheritance** the new class (called the subclass) starts with the characteristics of an existing class (called the superclass), possibly customizing them and adding unique characteristics of its own.
- e) **The Unified Modeling language (UML)** is a graphical language that allows people who design software systems to use an industry-standard notation to represent them.
- f) The size, shape, color and weight of an object are considered **attributes** of the object's class.

1.4 Fill in the blanks in each of the following statements:

- a) The logical unit that receives information from outside the computer for use by the computer is the **input unit**.
- b) The process of instructing the computer to solve a problem is called **programming**.
- c) **High level language** is a type of computer language that uses English like abbreviations for machine-language instructions.
- d) **Output unit** is a logical unit that sends information which has already been processed by the computer to various devices so that it may be used outside the computer.
- e) **Memory** and **storage** are logical units of the computer that retain information.
- f) is a logical unit of the computer that performs calculations.
- g) **CPU** is a logical unit of the computer that makes logical decisions.
- h) **High level language** languages are most convenient to the programmer for writing programs quickly and easily.
- i) The only language a computer can directly understand is that computer's **Machine language**.
- j) **CPU** is a logical unit of the computer that coordinates the activities of all the other logical units.

1.5 Fill in the blanks in each of the following statements:

- a) **Java** is a platform independent programming language that was built with the objective of allowing programs to be written once and then run on a large variety of electronic devices without modification.
- b) **Standard**, **Enterprise** and **Micro** are the names of the three editions of Java that can be used to build different kinds of applications.
- c) **Bandwidth** is the information-carrying capacity of communication lines, and has rapidly increased over the years and become more affordable. Its availability is a cornerstone for building applications that are significantly connected.
- d) An **assembler** is a translator that can convert early assembly-language programs to machine language with reasonable efficiency.

1.6 Fill in the blanks in each of the following statements:

- a) Java programs normally go through five phases **editing**, **compilation**, **loading**, **execution**, and **testing**.
- b) A(n) **IDE** provides many tools that support the software development process, such as editors for writing and editing programs, debuggers for locating logic errors in programs, and many other features.
- c) The command java invokes the **JVM**, which executes Java programs.
- d) A(n) **emulator** is a software application that simulates a computer, but hides the underlying operating system and hardware from the programs that interact with it.
- e) The **class loader** takes the .class files containing the program's bytecodes and transfers them to primary memory.
- f) The **bytecode verifier** examines bytecodes to ensure that they're valid.

1.7 Explain what a just-in-time (JIT) compiler of Java does.

Ans: A just-in-time (JIT) compiler of Java is a component of the Java virtual machine (JVM) that dynamically compiles bytecode into native machine code at runtime, just before the code is executed. This allows the Java code to run faster because it is translated into a form that can be more directly executed by the CPU.

When Java code is executed, it is initially interpreted by the JVM. This interpretation process can be relatively slow, because the JVM must repeatedly analyze and execute the same code each time the program is run. To improve performance, the JIT compiler optimizes commonly executed code by translating it into machine code and storing the compiled code in memory. The next time the code is executed, the JVM can directly execute the optimized machine code, which is typically much faster than interpreting the Java bytecode.

1.8 One of the world's most common objects is a wrist watch. Discuss how each of the following terms and concepts applies to the notion of a watch: object, attributes, behaviors, class, inheritance (consider, for example, an alarm clock), modeling, messages, encapsulation, interface and information hiding.

Object: A watch is an object that can be described as a timepiece that is worn on the wrist to keep track of time. It can be physical, digital or a combination of both.

Attributes: A watch has various attributes such as its material, color, shape, size, design, and the type of information it provides, such as time, date, alarms, and other features.

Behaviors: A watch has certain behaviors, such as displaying the time accurately, having the ability to set alarms, tracking time in different time zones, and other features that allow users to interact with it.

Class: A watch belongs to the class of timepieces, which include other objects that are used to measure or display time, such as clocks, timers, sundials and hourglasses.

Inheritance: Inheritance is the concept in object-oriented programming where a class can derive properties and attributes from another class. For example, an alarm clock is a type of watch that can inherit properties from the watch class.

Modeling: Modeling is the process of representing real-world objects or systems using object-oriented programming languages or other modeling languages. A watch can be modeled using these languages to create software simulations that replicate its behaviors and interactions with users.

Messages: In object-oriented programming, objects communicate with each other through messages. For example, if a user wants to set an alarm on a watch, they would send a message to the watch object informing it to set the alarm.

Encapsulation: Encapsulation is a mechanism in object-oriented programming that protects the internal workings of an object from external access. This ensures that the object's state and behavior are kept private and can only be accessed through defined interfaces. A watch can implement encapsulation by hiding its internal workings from users and only exposing certain methods and properties.

Interface: An interface is a set of methods and properties that define how an object can be used by other objects. A watch's interface could include methods for displaying the time, setting alarms and changing time zones, and properties such as the watch's physical size and color.

Information hiding: Information hiding is the practice of limiting access to an object's internal attributes and functionality to ensure that it is used correctly and efficiently. A watch can implement information hiding by only exposing certain attributes and methods to users, and hiding the details of how it functions internally.