Chapter 1 Introduction to Computers, Programs, and Java

Section 1.2 What is a Computer?

1. is the physical aspect of the computer that can be seen.
   1. Hardware
   2. Software
   3. Operating system
   4. Application program

Key:a See the first paragraph in Section 1.2.

#

Section 1.2.1 What is a Computer?

1. is the brain of a computer.
   1. Hardware
   2. CPU
   3. Memory
   4. Disk

Key:b See the first paragraph in Section 1.2.1.

#

1. The speed of the CPU may be measured in .
   1. megabytes
   2. gigabytes
   3. megahertz
   4. gigahertz

Key:cd See the third paragraph in Section 1.2.1. 1 megahertz equals 1 million pulses per second and 1 gigahertz is 1000 megahertz.

#

Section 1.2.2 Bits and Bytes

1. Why do computers use zeros and ones?
   1. because combinations of zeros and ones can represent any numbers and characters.
   2. because digital devices have two stable states and it is natural to use one state for 0 and the other for 1.
   3. because binary numbers are simplest.
   4. because binary numbers are the bases upon which all other number systems are built. Key:b See the second paragraph in Section 1.2.2.

#

1. One byte has bits.
   1. 4
   2. 8
   3. 12
   4. 16

Key:b See the thrid paragraph in Section 1.2.2.

#

1. One gigabyte is approximately bytes.
   1. 1 million
   2. 10 million
   3. 1 billion
   4. 1 trillion

Key:c See the fifth paragraph in Section 1.2.2.

#

Section 1.2.3 Memory

1. A program and its data must be moved into the computer's before they can be executed by the CPU.
   1. memory
   2. hard disk
   3. CPU
   4. CD-ROM

Key:a See the first paragraph in Section 1.2.3.

#

Section 1.2.4 Storage Devices

1. A computer's is volatile; that is, any information stored in it is lost when the system's power is turned off.
   1. memory
   2. hard disk
   3. flash stick
   4. CD-ROM

Key:a See the first paragraph in Section 1.2.4.

#

1. Which of the following are storage devices?
   1. portable disk
   2. hard disk
   3. flash stick
   4. CD-ROM

Key:abcd See the second paragraph in Section 1.2.4.

#

Section 1.2.5 Input and Output Devices

1. The specifies the number of pixels in horizontal and vertical dimensions of the display device.
   1. screen resolution
   2. pixel
   3. dot pitch
   4. monitor

Key:a See the third last paragraph in Section 1.2.5.

#

Section 1.2.6 Communications Devices

1. is a device to connect a computer to a local area network (LAN).
   1. Regular modem
   2. DSL
   3. Cable modem
   4. NIC

Key:d See the bullet items in Section 1.2.6.

#

Section 1.3 Program Languages

1. are instructions to the computer.
   1. Hardware
   2. Software
   3. Programs
   4. Keyboards

Key:bc See the Key Point in Section 1.3.

#

1. Computer can execute the code in .
   1. machine language
   2. assembly language
   3. high-level language
   4. none of the above

Key:a See the first paragraph in Section 1.3.1.

#

1. translates high-level language program into machine language program.
2. An assembler
3. A compiler
4. CPU
5. The operating system

Key:b See the first paragraph after Table 1.1 in Section 1.3.3.

#

Section 1.4 Operating Systems

1. is an operating system.
2. Java
3. C++
4. Windows
5. Visual Basic
6. Ada

Key:c See the first paragraph in Section 1.4.

#

1. is a program that runs on a computer to manage and control a computer's activities.
2. Operating system
3. Java
4. Modem
5. Interpreter
6. Compiler

Key:a See the first paragraph in Section 1.4.

#

Section 1.5 Java, World Wide Web, and Beyond

1. Due to security reasons, Java cannot run from a Web browser in the new version of Java.
2. applications
3. applets
4. servlets
5. Micro Edition programs

Key:b See the last paragraph in Section 1.5.

#

2. Java
3. C++
4. C
5. C#
6. Python

is not an object-oriented programming language.

Key:c See Table 1.1 in Section 1.3.1.

#



is interpreted.

1. Java
2. C++
3. C
4. Ada
5. Pascal

Key:a See the second paragraph in Section 1.5.

#

2. Java
3. C++
4. C
5. Ada
6. Pascal

is architecture-neutral.

Key:a See the second paragraph in Section 1.5.

#

Section 1.6 The Java Language Specification, API, JDK, and IDE



language.

is a technical definition of the language that includes the syntax and semantics of the Java programming

1. Java language specification
2. Java API
3. Java JDK
4. Java IDE

Key:a See the last paragraph in Section 1.6.

#

1. contains predefined classes and interfaces for developing Java programs.
2. Java language specification
3. Java API
4. Java JDK
5. Java IDE

Key:b See the last paragraph in Section 1.6.

#

1. consists of a set of separate programs for developing and testing Java programs, each of which is invoked from a command line.
2. Java language specification
3. Java API
4. Java JDK
5. Java IDE

Key:c See the last paragraph in Section 1.6.

#



provides an integrated development environment (IDE) for rapidly developing Java programs. Editing,

compiling, building, debugging, and online help are integrated in one graphical user interface.

1. Java language specification
2. Java API
3. Java JDK
4. Java IDE

Key:d See the last paragraph in Section 1.6.

#

Section 1.7 A Simple Java Program

1. The main method header is written as:
2. public static void main(string[] args)
3. public static void Main(String[] args)
4. public static void main(String[] args)
5. public static main(String[] args)
6. public void main(String[] args)

Key:c In A, string should be spelled String with S in uppercase. In B, Main should be spelled main in lowercase. In D, the keyword void is missing. in D, the keyword static is missing. C is correct.

#

1. Which of the following statements is correct?
2. Every line in a program must end with a semicolon.
3. Every statement in a program must end with a semicolon.
4. Every comment line must end with a semicolon.
5. Every method must end with a semicolon.
6. Every class must end with a semicolon.

Key:b Only statements need to be end with a semicolon.

#

1. Which of the following statements is correct to display Welcome to Java on the console?
2. System.out.println('Welcome to Java');
3. System.out.println("Welcome to Java");
4. System.println('Welcome to Java');
5. System.out.println('Welcome to Java");
6. System.out.println("Welcome to Java');

Key:b In A, a string must be enclosed in double quotation marks. In C, a string must be enclosed in double quotation marks and also the .out is missing. In D, a string must be enclosed in double quotation marks on both sides. In E, a string must be enclosed in double quotation marks. B is correct.

#

Section 1.8 Creating, Compiling, and Executing a Java Program

1. The JDK command to just compile a class (not run) in the file Test.java is
2. java Test
3. java Test.java
4. javac Test.java
5. javac Test
6. JAVAC Test.java

Key:c In A, java is the command to run a program. In B, java is the command to run a program. In D, java is the command to run a program. In E, The command is case sensitive. C is correct.

#

1. Which JDK command is correct to run a Java application in ByteCode.class?
2. java ByteCode
3. java ByteCode.class
4. javac ByteCode.java
5. javac ByteCode
6. JAVAC ByteCode

Key:a A is correct. In B, to compile, use only the class name, not including the extension. In C, javac is the command to compile a class, not to run a class. In D, javac is the command to compile a class, not to run a class. In E, the command is case sensitive and javac is the command to compile a class, not to run a class.

#

1. Java compiler translates Java source code into .
2. Java bytecode
3. machine code
4. assembly code
5. another high-level language code

Key:a Java bytecode is the end result of compiling a Java source file. Java bytecode is machine-indepedent and can be run on any platform.

#

1. is a software that interprets Java bytecode.
2. Java virtual machine
3. Java compiler
4. Java debugger
5. Java API

Key:a Java virtual machine is a software that runs Java bytecode. Java compiler compiles Java source code into Java bytecode.

#

1. Suppose you define a Java class as follows, the source code should be stored in a file named .

public class Test {

}

1. Test.class
2. Test.doc
3. Test.txt
4. Test.java
5. Any name with extension .java

Key:d You have to name the class as ClassName.java. Here the class name is Test.

#

1. The extension name of a Java bytecode file is
2. .java
3. .obj
4. .class
5. .exe

Key:c .java is the Java source code file name and .class is the Java bytecode file name.

#

1. The extension name of a Java source code file is
2. .java
3. .obj
4. .class
5. .exe

Key:a .java is the Java source code file name and .class is the Java bytecode file name.

#

1. Which of the following lines is not a Java comment?
2. /\*\* comments \*/
3. // comments
4. -- comments
5. /\* comments \*/
6. \*\* comments \*\*

Key:ce The text that begins with // in a line is known as a line comment and the text that is enclosed between /\* and \*/ is known as a paragraph comment.

#

1. Which of the following are the reserved words?
2. public
3. static
4. void
5. class

Key:abcd These are all reserved words with specific meaning to the compiler.

#

1. Every statement in Java ends with .
2. a semicolon (;)
3. a comma (,)
4. a period (.)
5. an asterisk (\*)

Key:a You have to tell the compiler where a statement ends using a semicolon.

#

1. A block is enclosed inside .
2. parentheses
3. braces
4. brackets
5. quotes

Key:b A block is enclosed inside a pair of openning and closing braces.

#

Section 1.9 Programming Style and Documentation

1. Programming style is important, because .
2. a program may not compile if it has a bad style
3. good programming style can make a program run faster
4. good programming style makes a program more readable
5. good programming style helps reduce programming errors

Key:cd Bad style is a readability issue. The code will still compile. Programming style is a readability issue and it does not effect the performance of the code. D and C are correct.

#

1. Analyze the following code.

I:

public class Test {

public static void main(String[] args) { System.out.println("Welcome to Java!");

}

}

II:

public class Test { public static void main(String[] args) { System.out.println("Welcome to Java!"); } }

1. Both I and II can compile and run and display Welcome to Java, but the code in II has a better style than I.
2. Only the code in I can compile and run and display Welcome to Java.
3. Only the code in II can compile and run and display Welcome to Java.
4. Both I and II can compile and run and display Welcome to Java, but the code in I has a better style than II. Key:d Both I and II are the same except the format is different. Format is a readability issue only.

#

1. Which of the following code has the best style?

I:

public class Test {

public static void main(String[] args) { System.out.println("Welcome to Java!");

}

}

II:

public class Test {

public static void main(String[] args) { System.out.println("Welcome to Java!");

}

}

III:

public class Test {

public static void main(String[] args) { System.out.println("Welcome to Java!");

}

}

IV:

public class Test {

public static void main(String[] args) { System.out.println("Welcome to Java!");

}

}

1. I
2. II
3. III
4. IV

Key:d All the code will run fine, but not formatted correctly. Only the code in IV is formatted correctly.

#

Section 1.10 Programming Errors

1. If a program compiles fine, but it produces incorrect result, then the program suffers .
2. a compile error
3. a runtime error
4. a logic error

Key:c If a program has a compile error, it will not run. When a program has a runtime error, it terminates abnormally. C is correct.

#

1. If you forget to put a closing quotation mark on a string, what kind of error will be raised?
2. a compile error
3. a runtime error
4. a logic error

Key:a This is a syntax error, which will be detected by the compiler.