Chapter 1: Introduction to Computers, the Internet and the Web

The company that popularized personal computing was _____. **Apple**

The computer that made personal computing legitimate in business and industry was the. **IBM**

Computers process data under the control of sets of instructions called computer _____.

Programs

The six key logical units of the computer are the _____, _____, _____, _____,

_____ and the _____. **input unit - output unit - memory unit - arithmetic and logic unit - central processing unit - secondary storage unit**

The three classes of languages discussed in the chapter are _____, _____, and _____.

machine languages - assembly languages - high-level languages

The programs that translate high-level language programs into machine language are called

_____. **Compilers**

The

_____ allows computer users to locate and view multimedia-based documents on

almost any subject over the Internet. **World Wide Web**

Allows a Java program to perform multiple activities in parallel**. Multithreading**

The _____ command from the JDK

executes a Java application. **Java**

The _____ command from the JDK compiles a Java program. **Java c**

A Java program file must end with file extension**. .java**

When a Java program is complied, the file produced by the complier ends with file extension.

.class

The file produced by the Java complier contains that are executed by the Java Virtual Machine. **Bytecodes**

Objects have the property of _____ although objects may know how to communicate with one another across well-defined interfaces, they normally are not allowed to know how other objects are implemented. **information hiding**

Java programmers concentrate on creating _____ which contain fields and the set methods that manipulate those fields and provide services to clients. **classes**

Classes can have relationships with other classes. **associations**

The process of analyzing and designing a system from an object oriented point of view is called _____. **object oriented analysis and design (OOAD)**

OOD also takes advantage of _____ relationships, where new classes of objects are derived by absorbing characteristics of existing classes, then adding unique characteristics of their own. **inheritance**

is a graphical language that allows people who design software systems to use an industry-standard notation to represent them. **Unified Modeling Language (UML)**

The size, shape, color and weight of an object are considered. **Attributes**

Categorize each of the following items as either hardware or software:

a) CPU **hardware**

b) compiler **software**

c) ALU **hardware**

d) interpreter **software**

e) input unit **hardware**

f) an editor program **software**

Which logical unit that receives information from outside the computer for use by the computer is the _____. **Input unit**

The process of instructing the computer to solve a problem is called _____. **Programming**

_____ is a type of computer language that uses English-like abbreviations for machine-language instructions. **Assembly language**

_____ 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. **Output unit**

_____ and _____ are logical units of the computer that retain information. **Memory unit and secondary storage unit**

_____ is a logical unit of the computer that performs calculations. **Arithmetic logic unit (ALU)**

_____ is a logical unit of the computer that makes logical decisions. **Logic unit**

_____ languages are most convenient to the programmer for writing programs quickly and easily. **High-level languages**

The only language a computer can directly understand is that computer's _____. **machine language**

_____ is a logical unit of the computer that coordinates the actives of all the other logical units. **Central processing unit (CPU)**

Fatal runtime errors cause program execution to stop while the non-fatal ones cause execution to finish, but with incorrect results.

The _____ programming language is now used to develop large-scale enterprise applications, to enhance the functionality of web servers, to provide applications for consumer devices and for many other purposes. **Java**

_____ initially became widely known as the development language of the UNIX operating system. **C**

Is a high-level language developed by John Kemeny and Thomas Kurtz at Dartmouth College in the mid 1960s. **BASIC (Beginner's All-purpose Symbolic Instruction Code)**

Is the oldest high-level language. Designed by John Backus for IBM during the late 1950s, **Fortran**

To develop commercial applications that require precise and efficient manipulation of data, we use. **COBOL**

The programming language developed by Bjarne Stroustrup in the early 1980s in b Bells Laboratories. **C++**

Java programs normally go through five phases to be executed - edit, compile, load, verify, and ___. **Execute**

A(n) _____ 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. **integrated development environments (IDEs)**

The command java invokes the _____, which executes Java programs. **JVM**

A(n) _____ is a software application that simulates a computer, but hides the underlying operating system and hardware from the programs that interact with it. **virtual machine**

A program that runs on multiple platforms. **Cross-platform software**

The _____ takes the .class files containing the program's bytecodes and transfers them to primary memory. **class loader**

The _____ examines bytecodes to ensure that they're valid. **bytecode verifier**

Chapter 2: Introduction to Java Applications

A __ begins the body of every method, and a __ ends the body of every method **{, }**

You can use the ___ statement to make decisions. **If**

_____ begins an end-of-line comment. **//**

Which of the following is NOT considered white space? **0**

_____ are reserved for use by Java. Keywords

Java applications begin execution at method: **main**

Which of the following methods display information in a command window.

System.out.print()

System.out.println()

System.out.printf()

all of the above

Comments cause the computer to print the text after the // on the screen when the program executes. **False**

All variables must be given a type when they're declared. true

Java considers the variables number and NuMbEr to be identical. **False**

The remainder operator (%) can be used only with integer operands. **False**

The arithmetic operators *, /, %, + and - all have the same level of precedence. **False**

Using proper java syntax declare variables c, thisIsAVariable, q76354 and number to be of type int.

int c;

int thisIsAVariable;

int q76354;

int number;

Prompt the user to enter an integer. **System.out.print("Enter an Integer:");**

Write a statement that would accept an integer input from the user and assign the result to a variable called value. Assume Scanner variable input can be used to read a value from the keyboard.

value = input.nextInt();;

Write a statement that would print "This is a Java program" on one line in the command window. Use method System.out.println.

System.out.println("This is a Java program");

Write a statement that would print "This is a Java program" on two lines in the command window. The first line should end with Java. Use System.out.printf and two %s format specifiers.

System.out.printf("%s%n%s%n", "This is a Java", "program";

Write a statement that would execute the following logic:

If the variable number is not equal to 7, display "The variable number is not equal to 7".

if (number != 7)

System.out.println("The variable number is not equal to 7");

Identify and correct the errors in the following statement:

if (c < 7);

System.out.println("c is less than 7");

if (c < 7) // semicolon removed

System.out.println("c is less than 7");

Identify and correct the errors in the following statement:

if (c => 7);

System.out.println("c is equal to or greater than 7");

if (c >= 7) // => corrected to >=

System.out.println("c is equal to or greater than 7");

Match the following operators to what operation they perform: *, /, %, +, -, =

multiplication, division, remainder, addition, subtraction, assignment

_____used to document a program and improve its readability. **Comments**

A decision can be made in a Java program with a(n) _____ **if statement**

Calculations are normally performed by_____ statements **Assignment statements**

The arithmetic operators with the same precedence as multiplication are_____

and_____ **/ division and % remainder**

When parentheses in an arithmetic expression are nested, the_____ set of parentheses

is evaluated first. **innermost**

A location in the computer's memory that may contain different values at various times throughout the execution of a program is called a(n) _____ **Variable**

Write a Java statement that accomplishes the following task:

Display the message "Enter an integer: ", leaving the cursor on the same line.

System.out.print("Enter an integer: ");

Assign the product of variables b and c to variable a.

a = b*c;

Use a comment to state that a program performs a sample payroll calculation.

//This program performs a sample payroll calculation.

Java operators are evaluated form left to right.

False. Some operators (Ex. Assignment (=)) are evaluated from right to left.

The following are all valid variable

names: **_under_bar_**, **m928134**, **t5**, **j7**, **her_sales$**, **his_$account_total**, **a**, **b$**, **c**, **z**, and **z2**.

True.

A valid Java arithmetic expression with no parentheses is evaluated from left to right.

False. The expression is evaluated using operator precedence.

Assuming that x = 2 and y = 3, what does the following statement display?

System.out.printf("x = %d%n", x);

x = 2

System.out.println("x = ");

x =

System.out.printf("Value of %d + %d is %d%n: , x, x, (x+ x));

Value of 2 + 2 is 4

System.out.printf("%d = %d%n", (x + y), (y + x));

5 = 5

Which of the following Java statements contain variables whose values are modified?

a) p = i + j + k + 7;

b) System.out.println("variables whose values are modified");

c) System.out.println("a = 5");

d) value = input.nextInt();

Statements (a) and (d) contain variables whose values have been modified.

Given that y = ax^3 +7, which of the following are correct Java statements for this equation?

a) y = a**x**x*x+7;

b) y = a**x**x*(x+7);

c) y = (a**x)**x*(x+7);

d) y = (a**x)**x*x+7;

e) y = a**(x**x*x)+7;

f) y = a**x**(x*x+7);

(a), (d), and (e) are correct Java statements from the given data.

State the order of evaluation of the operators in each of the following Java statements, and show the value of x after the statement is performed:

x = 7 + 3 * 6 / 2 - 1;

The order of evaluation operators are: *, /, +, -.

The value of x = 7+3*6/2-1

x= 7+ 18/2-1

x=7+9-1

x=16-1

The value of x = 15

State the order of evaluation of the operators in each of the following Java statements, and show the value of x after the statement is performed:

x = 2 % 2 + 2 * 2 -2 / 2;

The order of operators are %, *, /, +, -.

The value of x = 2%2+2*2-2/2

x=0+2*2-2/2

x=0+4-2/2

x=0+4-1

x=4-1

The value of x=3

State the order of evaluation of the operators in each of the following Java statements, and show the value of x after the statement is performed:

x = (3 **9** (3*9 / (3))));

The order of evaluation operators are ( ), *, /, +.

The value of x = (39(3*9/(3))));

x = (3*9(3+(27/(3))));

x = (39(3+9));

x = 39(12));

x = 27*12);

The value of x = 324

Write an application that displays the numbers 1 to 4 on the same line, with each pair of adjacent numbers separated by one space. Write the application using the following techniques:

a. Use one System.out.println statement.

b. Use four System.out.print statements.

c. Use one System. out. print statement.

```
// filename: Printing.java
public class Printing {
public static void main(String[] args) {
int num1 = 1;
int num2 = 2;
int num3 = 3;
int num4 = 4;
System.out.println(num1 + " " + num2 + " " + num3 + " " + num4);

System.out.print(num1 + " " + num2 + " " + num3 + " " + num4);

System.out.printf("\n%d %d %d %d",num1,num2,num3,num4); } }
```

Chapter 3: Introduction to Classes and Objects

A house is to a blueprint as a(n) _____ is to a class. **Object**

Keyword ___ in a class declaration is followed immediately by the class's name **class**

Each class declaration that begins with keyword ___ must be stored in a file that has exactly the same as the class and ends with the .javafile-name extension. **Public**

Keyword ___ requests memory from the system to store an object , then calls the corresponding class's constructor to initialize the object **new**

Each parameter must specify both a(n) ___ and a(n) ___ **type and name**

By default, classes that are compiled in the same directory are considered to be in the same package, known as the ___ **default package**

When each object of a class maintains its own copy of an attribute, the field that represents the attribute is also known as a(n) ___ **instance variable**

Java provides two primitive types for storing floating-point numbers in memory: ___ and ___ **float and double**

Variables of type double represent ___ floating-point numbers **double-precision**

Scanner method ___ reads characters until it encounters a newline character, then returns those characters as a String. **nextLine**

Scanner method ___ returns a double value. **nextDouble**

Keyword public is an access **modifier**

Return type ___ indicates that a method will not return a value void

Class String is in package **java.lang**

A(n) ___ is not required if you always refer to a class with its fully qualified class name **import declaration**

A(n) ___ is a number with a decimal point, such as 7.33, 0.0975 or 1000.12345 **floating-point number**

Variables of type float represent ___ floating-point numbers. **single-precision**

The format specifier ___ is used to output values of type float or double. **%f**

Types in Java are divided into two categories- ___ types and ___ types **Primitive and reference**

By convention, method names begin with an uppercase first letter, and all subsequent words in the name begin with a capital first letter

False; begins with lowercase first letter than all uppercase first letters

An import declaration is not required when one class in a package uses another in the same package

True

Empty parentheses following a method name in a method declaration indicate that the method does not require any parameters to perform its task.

True

Variables or methods declared with access modifier private are accessible only to methods of the class in which they're declared

True

A primitive-type variable can be used to invoke a method

false; a reference to an object is required to invoke the object's methods.

Variables declared in the body of a particular method are known as instance variables and can be used in all methods of the class

False; such variables are called local variables and can be used only if he method in which they're declared

Every method's body is delimited by left and right braces

True

Primitive-type local variables are initialized by default

False; Primitive-type instance variables are initialized by default. each local variable must explicitly be assigned a value

Reference-type instance variables are initialized by default to the value null

True

Any class that contains public static void main(String[] args) can be used to execute an application

True

The number of arguments in the method call must match the number of parameters in the method declaration's parameter list

True

Floating-point values that appear in the source code are known as floating-point literals and are type float by default.

False; Such literals are of type double by default

What is a difference between a local variable and a field?

A local variable is declared in the body of a method and can be used only from the point at which it's declared through the end of the method declaration. A field is declared in a class, but not in the body of any of the class's methods. Also, fields are accessible to all the methods of the class.

Explain the purpose of a method parameter. What is the difference between a parameter and an argument?

A parameter represents additional information that a method requires to perform its task. Each parameter required by a method is specified in the method's declaration . An argument is the actual value for a method parameter. When a method is called, the argument values are passed to the corresponding parameters of the method so that is can perform its task.

Chapter 4: Control Statements Part 1

All programs can be written in terms of three types of control structures **selection, sequence, and repetition**

The statement is used to execute one action when a condition is true and another when that condition is false **if else**

Repeating a set of instructions a specific number of times is called _____ repetition **definite**

When it's not known in advance how many times a set of statements will be repeated, a(n) _____ value can be used to terminate the repetition **dummy**

The _____ structure is built into Java; by default, statements execute in the order they appear **sequence**

Instance variables of types char, byte, short, int, long, float and double are all given the value _____ by default **zero**

Java is a(n) _____ language; it requires all variables to have a type **strongly typed**

If the increment operator is _____ to a variable, first the variable is incremented by 1, then its new value is used in the expression **prefixed**

An algorithm is a procedure for solving a problem in terms of the actions to execute and the order in which they execute. **True**

A set of statements contained within a pair of parentheses is called a block. **False. (A set of statements contained within a pair of braces ({ and }) is called a block.)**

A selection statement specifies that an action is to be repeated while some condition remains true **False. (A repetition statement specifies that an action is to be repeated while some condition remains true)**

A nested control statement appears in the body of another control statement true

Java provides the arithmetic compound assignment operators +=, -=, *=, /= and %= for abbreviating assignment expressions **true**

The primitive types (Boolean, char, byte, short, int, long, float and double) are portable across only Windows platforms **False. (The primitive types (Boolean, char, byte, short, int, long, float and double) are portable across all computer platforms that support Java)**

Specifying the order in which statements execute in a program is called program control **true**

The unary cast operator (double) creates a temporary integer copy of its operand **False. (The unary cast operator (double) creates a temporary floating-point copy of its operand)**

Pseudocode helps you think out a program before attempting to write it in a programming language **true**

Write Java statements to accomplish the following task: Use one statement to assign the sum of x and y to z, then increment x by 1

z = y + x++ (z = x++ + y;)

Write Java statements to accomplish the following task: Test whether variable count is greater than 10. If it is, print "Count is greater than 10".

```
if ( count > 10 )
System.out.println( "Count is greater than 10" );
```

Write Java statements to accomplish the following task: Use one statement to decrement the variable x by 1, then subtract it from variable total and store the result in variable total **total -= --x;**

Write Java statements to accomplish the following task: Calculate the remainder after q is divided by divisor, and assign the result to q. Write this statement in two different ways

q %= divisor and ; q = q % divisor;

Write Java statements to accomplish the following task: Declare variables sum and x to be of type int **int sum; int x;**

Write Java statements to accomplish the following task: Assign 1 to variable x **x = 1;**

Write Java statements to accomplish the following task: Assign 0 to variable sum **sum = 0;**

Write Java statements to accomplish the following task: Add variable x to variable sum, and assign the result to variable sum

sum += x; or sum = sum + x;

Write Java statements to accomplish the following task: Print "The sum is: ", followed by the value of variable sum **System.out.printf("The sum is: %d\n", sum);**

Identify and correct the errors in each of the following sets of code:

a) while (c <= 5)

{

product *= c;

++c;

Add a closing right brace after the statement ++c

Error: The closing right brace of the while statement's body is missing.

if (gender == 1)

System.out.println("Woman");

else;

System.out.println("Man");

Remove the semicolon after else Error: The semicolon after else results in a logic error. The second output statement will always be executed.

Chapter 5: Control Statements Part 2

Typically, _____ statements are used for counter-controlled repetition and _____ statements for sentinel-controlled repetition. **for, while**

The do...while statement tests the loop-continuation condition _____ executing the loop's body; therefore, the body always executes at least once. **after**

The _____ statement selects among multiple actions based on the possible values of an integer variable or expression, or a **String**

The _____ statement, when executed in a repetition statement, skips the remaining statements in the loop body and proceeds with the next iteration of the loop. **continue**

The ___ operator can be used to ensure that two conditions are both true before choosing a certain path of execution. **&&**

If the loop-continuation condition in a for loop header is initially _____, the program does not execute the for statement's body. **False**

Methods that perform common tasks and do not require objects are called _____ methods **static**

The default case is required in the switch selection statement. **False. The default case is optional.**

The break statement is required in the last case of a switch selection statement. **False. The break statement is used to exit the switch statement.**

The expression ((x > y) && (a > b)) is true if either or both of its operands are true. **False. Both operands must be true.**

An expression containing the II operator is true if either or both of its operands are true. **True**

The comma (.) formatting flag in a format specifier (e.g, % 20.2f) indicates that a value should be output with a thousands separator. **True**

To test for a range of values in a switch statement, use a hyphen (-) between the start and end values of the range in the case label. **False. Every value should be listed in a separate case label.**

Listing cases consecutively with no statements between them enables the cases to perform the same set of statements. **True**

Write a java statement that calculates and stores (in a variable named result), the value of 2.5 raised to the power of 3, using the pow method.

double result = Math.pow (2.5,3);

Match the following format specifiers with their proper type.

%s, String,

%d, integer

%f, floating point

%n. new line

Chapter 6: Methods Deeper Look

A method is invoked with a(n) _____. **method call**

A variable known only within the method in which it's declared is called a(n) _____.

local variable

The _____ statement in a called method can be used to pass the value of an expression back to the calling method. **Return**

The keyword _____ indicates that a method does not return a value. **void**

Data can be added or removed only from the _____ of a stack. **top**

Stacks are known as _____ data structures; the last item pushed (inserted) onto the stack is the first item popped (removed) from the stack. **LIFO (last-in, first-out)**

The three ways to return control from a called method to a caller are _____, _____, and _____. return, **return expression, closing right brace of method**

An object of class _____ produces truly random numbers. **Secure-Random**

The method-call stack contains the memory for local variables on each invocation of a method during a program's execution. This data, stored as a portion of the method-call stack, is known as the _____ or _____ of the method call. **stack frame, activation record**

If there are more method calls than can be stored on the method-call stack, an error known as a(n) _____ occurs. **stack overflow**

The _____ of a declaration is the portion of a program that can refer to the entity in the declaration by name. **scope**

It's possible to have several methods with the same name, each operating on different types or numbers of arguments. This feature is called method _____. **Overloading**

The program execution stack is also referred to as the _____ stack. **Method Call**

Chapter 8: Classes and Objects: A Deeper Look

When compiling a class in a package, the javac command-line option _____ specifies where to store the package and causes the compiler to create the package's directories if they do not exist. **-d**

String class static method _____ is similar to method System.out.printf, but returns a formatted String rather than displaying a String in a command window. **String.format()**

If a method contains a local variable with the same name as one of its class's fields, the local variable _____ the field in that method's scope. **Shadows**

The _____ method is called by the garbage collector just before it reclaims an object's memory. **finalize**

A(n) _____ declaration specifies one class to import. **single-type-import**

If a class declares constructors, the compiler will not create a(n) _____. **default constructor**

An object's _____ method is called implicitly when an object appears in code where a String is needed. **toString**

Get methods are commonly called _____ or _____. **accessor methods; query methods**

A(n) _____ method tests whether a condition is true or false. **predicate**

For every enum, the compiler generates a static method called _____ array of the enum's constants in the order in which they were declared. **Values**

Composition is sometimes referred to as a(n) _____ relationship. **Has-a**

A(n) _____ declaration contains a comma-separated list of constants. **Enum**

A(n) _____ variable represents classwide information that's shared by all the objects of the class. **static**

A(n) _____ declaration imports one static member. **single static import**

The _____ states that code should be granted only the amount of privilege and access that it needs to accomplish its designated task. **principle of least privilege**

Keyword _____ specifies that a variable is not modifiable. final

There can be only one _____ in a Java source-code file, and it must precede all other declarations and statements in the file. **package declaration**

A(n) _____ declaration imports only the classes that the program uses from a particular package. **type-import-on-demand**

The compiler uses a(n) _____ to locate the classes it needs in the classpath. **class loader**

The classpath for the compiler and JVM can be specified with the _____ option to the javac or java command, or by setting the _____ environment variable.

-classpath, CLASSPATH

Set methods are commonly called _____ because they typically change a value. **mutator methods**

A(n) _____ imports all static members of a class. **static import on demand**

The public methods of a class are also known as the class's _____ or _____. **public services; public interfaces**

Chapter 9: Object-Oriented Programming: Inheritance

___ is a form of software reusability in which new classes acquire the members of existing classes and embellish those classes with new capabilities. **Inheritance**

A superclass's _____ members can be accessed in the superclass declaration and in subclass declarations. **public and protected**

In a _____ relationship, an object of a subclass can also be treated as an object of its superclass. **is-a or inheritance**

In a _____ relationship, a class object has references to objects of other classes as members. **has-a or composition**

In single inheritance, a class exists in a _____ relationship with its subclasses. **Hierarchical**

A superclass's _____ members are accessible anywhere that the program has a reference to an object of that superclass or to an object of one of its subclasses. **Public**

When an object of a subclass is instantiated, a superclass _____ is called implicitly or explicitly. **constructor**

Subclass constructors can call superclass constructors via the _____ keyword. **Super**

Superclass constructors are not inherited by subclasses. **True**

A has-a relationship is implemented via inheritance. **False**

A Car class has an is-a relationship with the SteeringWheel and Brakes classes. **False**

When a subclass redefines a superclass method by using the same signature, the subclass is said to overload that superclass method. **False**