

Chapter 2 Introduction to Java Applications

Section 2.2 Your First Program in Java: Printing a Line of Text

2.2 Q1: End-of-line comments that should be ignored by the compiler are denoted using

- Two forward slashes (//).
- Three forward slashes (///).
- A slash and a star (/*).
- A slash and two stars (/**).
- ANS:** a. Two forward slashes (//).

2.2 Q2: Which of the following is *not* a valid Java identifier?

- my Value
- \$_AAA1
- width
- m_x

ANS: a. my Value (Identifiers may not contain blanks).

2.2 Q3: Which of the following *cannot* cause a syntax error to be reported by the Java compiler?

- Mismatched {}
- Missing */ in a comment that begins with /*
- Missing ;
- An extra blank line.

ANS: d. Extra blank lines.

2.2 Q4: Which of the following does *not* contain a syntax error?

- System.out.println('Hello world!');
- System.out.println("Hello
world!");
- System.out.println("Hello world!");
- System.out.println(Hello world!);

ANS: c. System.out.println("Hello world!");

Compiling and Executing Your First Java Application

2.2 Q5: Which command compiles the Java source code file `Welcome.java`?

- cd Welcome.java
- javac Welcome.java
- java Welcome.java
- compile Welcome.java

ANS: b. javac Welcome.java

2.2 Q6: Which command executes the Java class file `Welcome.class`?

- javac Welcome.class
- java Welcome.class
- java Welcome
- run Welcome.class

ANS: c. java Welcome

Section 2.3 Modifying Your First Java Program

Displaying a Single Line of Text with Multiple Statements

2.3 Q1: Which is the output of the following statements?

- ```
System.out.print("Hello ");
System.out.println("World");
```
- a. Hello World
  - b. HelloWorld
  - c. Hello  
World
  - d. World  
Hello

**ANS:** a. Hello World

### *Displaying Multiple Lines of Text with a Single Statement*

2.3 Q2: Which of the following is the *escape character*?

- a. \*
- b. \
- c. \n
- d. "

**ANS:** b. \

2.3 Q3: Which of the following statements will print a single line containing "hello there"?

- a. System.out.println( "hello" );  
System.out.println( " there" );
- b. System.out.println( "hello" , " there" );
- c. System.out.println( "hello" );  
System.out.print( " there" );
- d. System.out.print( "hello" );  
System.out.println( " there" );

**ANS:** d. System.out.print( "hello" );  
System.out.println( " there" );

2.3 Q4: Which of the following escape sequences represents a *carriage return*?

- a. \n.
- b. \r.
- c. \cr.
- d. \c.

**ANS:** b. \r.

2.3 Q5: Which of the following statements would display the phrase Java is fun?

- a. System.out.println( "hellois fun\rJava " );
- b. System.out.println( 'Java is fun' );
- c. System.out.println( "\"Java is fun\"" );
- d. System.out.println( Java is fun );

**ANS:** a. System.out.println( "hellois fun\rJava " );

## Section 2.4 Displaying Text with `printf`

2.4 Q1: When method `printf` requires multiple arguments, the arguments are separated with \_\_\_\_\_.

- a. colons (:).
- b. semicolons (;).
- c. commas (,).
- d. periods (.

**ANS:** c. commas (,).

2.4 Q2: Which of the following statement displays `Hello World`?

- a. `System.out.printf( "%2s", "Hello " "World" );`
- b. `System.out.printf( "%s %s", "Hello", "World" );`
- c. `System.out.printf( "%s%s", "Hello, World" );`
- d. `System.out.printf( "s% s%", "Hello", "World" );`

**ANS:** b. `System.out.printf( "%s %s", "Hello", "World" );`

## Section 2.5 Another Application: Adding Integers

2.5 Q1: All `import` declarations *must* be placed

- a. inside the class declaration's body.
- b. before the class declaration.
- c. after the class declaration.
- d. all of the above will work.

**ANS:** b. before the class declaration.

2.5 Q2: Which of the following is a variable declaration statement?

- a. `int total;`
- b. `import java.util.Scanner;`
- c. `public static void main( String args[] )`
- d. `// first string entered by user`

**ANS:** a. `int total;`

2.5 Q3: A(n) \_\_\_\_\_ enables a program to read data from the user.

- a. `printf`.
- b. `import` declaration.
- c. `Scanner`.
- d. `main`.

**ANS:** c. `Scanner`.

2.5 Q4: Which of the following is *not* a Java primitive type?

- a. `char`
- b. `byte`
- c. `real`
- d. `double`

**ANS:** c. `real`

2.5 Q5: The format specifier \_\_\_\_\_ is a placeholder for an `int` value?

- a. `%a`
- b. `%d`
- c. `%int`
- d. `%s`

**ANS:** b. `%d`

## Section 2.6 Memory Concepts

2.6 Q1: Which of the following statements does *not* alter a memory location?

- a. `int a;`
- b. `number = 12;`
- c. `y = y + 2;`
- d. `width = Integer.parseInt(input);`

**ANS:** a. `int a;`

## Section 2.7 Arithmetic

2.7 Q1: What is the value of `result` after the following Java statements execute?

```
int a, b, c, d, result;
a = 4;
b = 12;
c = 37;
d = 51;
result = d % a * c + a % b + a;
```

- a. 119
- b. 51
- c. 127
- d. 59

**ANS:** a. 119

2.7 Q2: Which of the following is *not* an arithmetic operator?

- a. `+`
- b. `-`
- c. `.`
- d. `%`

**ANS:** c. `.`

## Section 2.8 Decision Making: Equality and Relational Operators

2.8 Q1: What will be output after the following Java statements have been executed?

```
int a, b, c, d;
a = 4;
b = 12;
c = 37;
d = 51;

if (a < b)
 System.out.println("a < b");

if (a > b)
 System.out.println("a > b");

if (d <= c)
 System.out.println("d <= c");

if (c != d)
 System.out.println("c != d");
```

- a. a < b  
c != d
- b. a < b  
d <= c  
c != d
- c. a > b  
c != d
- d. a < b  
c < d  
a != b

**ANS:** a. a < b  
c != d

2.8 Q2: Which of the following is *not* a compilation error?

- a. Neglecting to initialize a local variable in a method before it is used.
- b. Placing a semicolon at the end of the first line of an `if` statement.
- c. Omitting the left and right parenthesis for the condition of an `if` statement.
- d. All are compilation errors.

**ANS:** b. Placing a semicolon at the end of the first line of an `if` statement.

2.8 Q3: Each of the following is a relational or equality operator except:

- a. <=
- b. !=
- c. ==
- d. >

**ANS:** b. !=