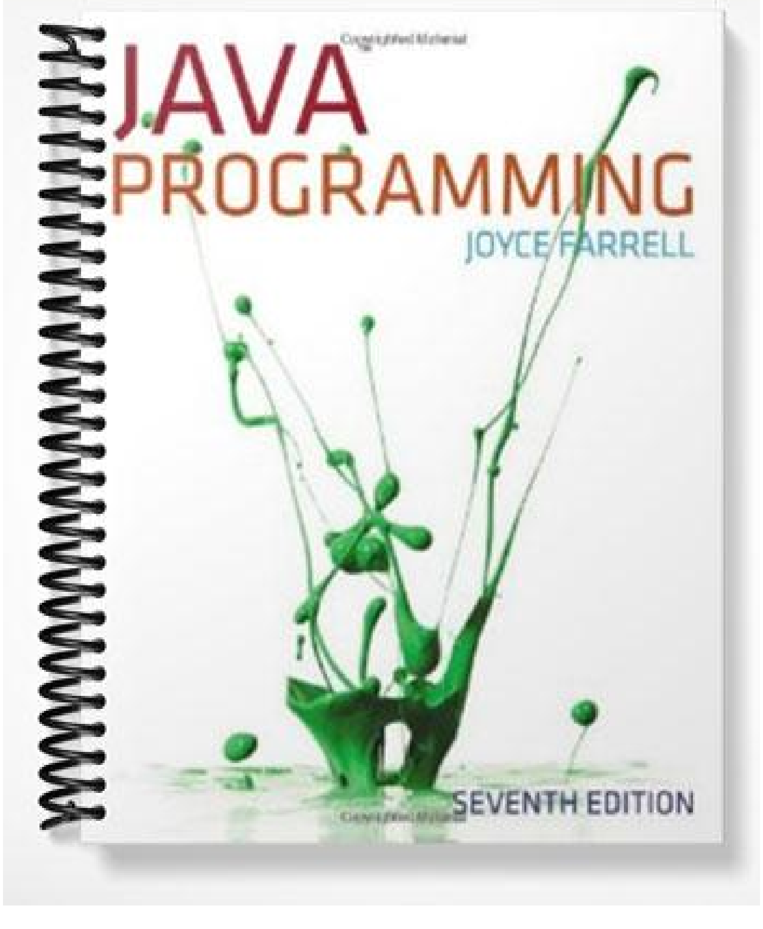
TEST BANK


Copyrighted Material
JAVA
PROGRAMMING
JOYCE FARRELL
SEVENTH EDITION

Chapter 2: Using Data

TRUE/FALSE

1. A variable can hold more than one value at a time.

   ANS: F PTS: 1 REF: 52

2. The legal integer values are -2^{31} through 2^{31}-1. These are the highest and lowest values that you can store in four bytes of memory, which is the size of an `int` variable.

   ANS: T PTS: 1 REF: 62

3. Multiplication, division, and remainder always take place after addition or subtraction in an expression.

   ANS: F PTS: 1 REF: 93

4. The term *parse* means to break into component parts.

   ANS: T PTS: 1 REF: 108

5. You can create a confirm dialog box with five arguments.

   ANS: T PTS: 1 REF: 90

6. Once a variable has been declared and initialized, new values may not be assigned to the variable.

   ANS: F PTS: 1 REF: 53

7. The expression `boolean isTenLarger = (10 < 5)` will produce a value of `true`.

   ANS: F PTS: 1 REF: 68

8. Even if a statement occupies multiple lines, the statement is not complete until the semicolon is reached.

   ANS: T PTS: 1 REF: 54

9. You are limited to declaring a maximum of three variables in a single statement.

   ANS: F PTS: 1 REF: 54

10. Constants hold a single value for the duration of the program execution.

    ANS: T PTS: 1 REF: 58

MULTIPLE CHOICE

1. A data item is ____ when it cannot be changed while a program is running.
   a. variable c. primitive

b. constant | d. literal

ANS: B | PTS: 1 | REF: 52

2. A ____ is a named memory location that you can use to store a value.
   a. cast | c. reference
   b. variable | d. primitive

   ANS: B | PTS: 1 | REF: 52

3. Primitive types serve as the building blocks for more complex data types, called ____ types.
   a. integer | c. reference
   b. literal | d. data

   ANS: C | PTS: 1 | REF: 52

4. ____ refers to the order in which values are used with operators.
   a. Associativity | c. Declaration
   b. Initialization | d. Floating

   ANS: A | PTS: 1 | REF: 53

5. In Java, you use variables of type ____ to store integers, or whole numbers.
   a. `num` | c. `var`
   b. `double` | d. `int`

   ANS: D | PTS: 1 | REF: 52

6. A(n) ____ variable can hold only one of two values: `true` or `false`.
   a. `integer` | c. `true`
   b. `boolean` | d. `comparison`

   ANS: B | PTS: 1 | REF: 67

7. The term ____ refers to the mathematical accuracy of a value.
   a. float data | c. significant digits
   b. real integers | d. single-precision floating-point number

   ANS: C | PTS: 1 | REF: 69

8. A ____ data type can hold 14 or 15 significant digits of accuracy.
   a. `double` | c. `char`
   b. `float` | d. `boolean`

   ANS: A | PTS: 1 | REF: 69

9. You use the ____ data type to hold any single character.
   a. `single` | c. `byte`
   b. `char` | d. `float`

   ANS: B | PTS: 1 | REF: 70

10. In Java, ____ is a built-in class that provides you with the means for storing and manipulating character strings.
    a. `Escape` | c. `String`
    b. `Type` | d. `Character`

ANS: C PTS: 1 REF: 72

11. You can store any character, including nonprinting characters such as a backspace or a tab, in a(n) ____ variable.
a. `int`
b. `char`
c. `boolean`
d. `set`

ANS: B PTS: 1 REF: 73

12. The characters ____ move the cursor to the next line when used within a `println()` statement.
a. `/n`
b. `\n`
c. `.+`
d. `$`

ANS: B PTS: 1 REF: 73-74

13. In Java, when a numeric variable is concatenated to a `String` using the ____, the entire expression becomes a `String`.
a. plus sign
b. equal sign
c. concatenate statement
d. string statement

ANS: A PTS: 1 REF: 56

14. You use ____ operators to perform calculations with values in your programs.
a. calculation
b. arithmetic
c. integer
d. precedence

ANS: B PTS: 1 REF: 91

15. ____ occurs when both of the operands are integers.
a. Data modeling
b. Type cast
c. Integer division
d. Unlike assignment

ANS: C PTS: 1 REF: 92

16. The percent sign is the ____ operator.
a. remainder
b. remaining
c. percentage
d. integer division

ANS: A PTS: 1 REF: 92

17. What is the value of `result` after the following statement is executed?
```
int result = 2 + 3 * 4;
```
a. 9
b. 10
c. 14
d. 20

ANS: C PTS: 1 REF: 93

18. The ____ is the type to which all operands in an expression are converted so that they are compatible with each other.
a. unifying type
b. data type
c. numbered
d. primitive

ANS: A PTS: 1 REF: 99

19. A(n) ____ dialog box asks a question and provides a text field in which the user can enter a response.
a. question
c. confirm

b. `JOptPane` d. input

ANS: D PTS: 1 REF: 85

20. Each primitive type in Java has a corresponding class contained in the `java.lang` package. These classes are called ____ classes.
    a. case c. type-wrapper
    b. primitive d. show

    ANS: C PTS: 1 REF: 87-88

21. A(n) ____ dialog box displays the options Yes, No, and Cancel.
    a. confirm c. message
    b. input d. answer

    ANS: A PTS: 1 REF: 89

22. Which of the following is NOT a component of a variable declaration statement?
    a. data type identifier c. variable name
    b. symbolic constant d. ending semicolon

    ANS: B PTS: 1 REF: 53

23. You may declare an unlimited number of variables in a statement as long as the variables are ____.
    a. the same data type c. properly commented
    b. initialized to the same value d. floating point numbers

    ANS: A PTS: 1 REF: 54

24. When a numeric variable is concatenated to a `String`, the entire expression becomes a(n) _____.
    a. `int` c. method
    b. constant d. `String`

    ANS: D PTS: 1 REF: 56

25. Which escape sequence will move the cursor to the beginning of the current line?
    a. `\b` c. `\\`
    b. `\r` d. `\n`

    ANS: B PTS: 1 REF: 73

COMPLETION

1. A(n) ____________________ is a simple data type.

   ANS: primitive type

   PTS: 1 REF: 52

2. A(n) ____________________ operator compares two items and the result has a Boolean value.

   ANS:
   relational
   comparison

PTS: 1 REF: 68

3. A(n) ____________________ number contains decimal positions.

ANS:
floating-point
`float`
`double`

PTS: 1 REF: 69

4. ____________________ forces a value of one data type to be used as a value of another type.

ANS:
Type casting
type casting
Casting
casting

PTS: 1 REF: 100

5. When you write programs that accept ____________________, there is a risk that the user will enter the wrong type of data.

ANS: user input

PTS: 1 REF: 81

MATCHING

Match each term with the correct statement below.

a.	operand	f.	primitive
b.	cast operator	g.	`float`
c.	assignment	h.	`boolean`
d.	operator precedence	i.	escape sequence
e.	garbage		

1. `true` or `false`
2. The operator that is represented by an equal sign (=)
3. A programming term for an unknown value
4. Java consistently specifies their size and format
5. A value that can be used on either side of an operator
6. Rules for the order in which parts of a mathematical expression are evaluated
7. A floating-point data type
8. Created by placing the desired result type in parentheses
9. Begins with a backslash followed by a character

1.	ANS: H	PTS: 1	REF: 52
2.	ANS: C	PTS: 1	REF: 53
3.	ANS: E	PTS: 1	REF: 54
4.	ANS: F	PTS: 1	REF: 62
5.	ANS: A	PTS: 1	REF: 91

6.	ANS: D	PTS: 1	REF: 93
7.	ANS: G	PTS: 1	REF: 69
8.	ANS: B	PTS: 1	REF: 100
9.	ANS: I	PTS: 1	REF: 73

SHORT ANSWER

1. A variable declaration is a statement that reserves a named memory location. It includes what four elements?

   ANS:
   A data type that identifies the type of data that the variable will store
   An identifier that is the variable’s name
   An optional assignment operator and assigned value, if you want a variable to contain an initial value
   An ending semicolon

   PTS: 1 REF: 53

2. Describe the variation types `byte`, `short`, and `long` of the integer type.

   ANS:
   The types `byte`, `short`, and `long` are all variations of the integer type. The `byte` and `short` types occupy less memory and can hold only smaller values; the `long` type occupies more memory and can hold larger values.

   PTS: 1 REF: 62

3. Describe how to assign values based on the result of comparisons to Boolean variables.

   ANS:
   Java supports six relational operators that are used to make comparisons. A relational operator compares two items; an expression that contains a relational operator has a Boolean value. When you use any of the operators that have two symbols (==, <=, >=, or !=), you cannot place any whitespace between the two symbols. You also cannot reverse the order of the symbols. That is, =<, =>, and =! are all invalid operators.

   PTS: 1 REF: 68

4. What is the difference between the `float` data type and the `double` data type?

   ANS:
   Java supports two floating-point data types: `float` and `double`. A `float` data type can hold floating-point values of up to six or seven significant digits of accuracy. A `double` data type requires more memory than a `float`, and can hold 14 or 15 significant digits of accuracy. The term *significant digits* refers to the mathematical accuracy of a value. For example, a `float` given the value 0.324616777 displays as 0.324617 because the value is accurate only to the sixth decimal position.

   PTS: 1 REF: 69

5. What is an escape sequence and why would a Java programmer use it to store a character?

   ANS:

You can store any character—including nonprinting characters such as a backspace or a tab—in a `char` variable. To store these characters, you can use an escape sequence, which always begins with a backslash followed by a character—the pair represents a single character.

PTS: 1 REF: 73

6. Describe and give an example of operator precedence.

   ANS:
   Operator precedence refers to the rules for the order in which parts of a mathematical expression are evaluated. The multiplication, division, and remainder operators have the same precedence. Their precedence is higher than that for the addition and subtraction operators. Addition and subtraction have the same precedence. In other words, multiplication, division, and remainder always take place from left to right prior to addition or subtraction in an expression. For example, the following statement assigns 14 to `result`: `int result = 2 + 3 * 4;`.

   PTS: 1 REF: 93

7. In Java, how is it possible to perform mathematical operations on operands with unlike types?

   ANS:
   When you perform arithmetic operations with operands of unlike types, Java chooses a unifying type for the result. The unifying type is the type to which all operands in an expression are converted so that they are compatible with each other. Java performs an implicit conversion; that is, it automatically converts nonconforming operands to the unifying type.

   PTS: 1 REF: 99

8. Explain how you can override a unifying type imposed by Java.

   ANS:
   You can explicitly (or purposely) override the unifying type imposed by Java by performing a type cast. Type casting forces a value of one data type to be used as a value of another type. To perform a type cast, you use a cast operator, which is created by placing the desired result type in parentheses. Using a cast operator is an explicit conversion. The cast operator is followed by the variable or constant to be cast.

   PTS: 1 REF: 100

9. How can you create and use an input dialog box in Java?

   ANS:
   You can create an input dialog box using the `showInputDialog()` method. Six overloaded versions of this method are available, but the simplest version uses a single argument that is the prompt you want to display within the dialog box. The `showInputDialog()` method returns a `String` that represents a user's response; this means that you can assign the `showInputDialog()` method to a `String` variable and the variable will hold the value that the user enters.

   PTS: 1 REF: 85-86

10. How would you ask the user to confirm an action using a dialog box?

ANS:
A confirm dialog box displays the options Yes, No, and Cancel; you can create one using the `showConfirmDialog()` method in the `JOptionPane` class. Four overloaded versions of the method are available; the simplest requires a parent component (which can be `null`) and the `String` prompt that is displayed in the box. The `showConfirmDialog()` method returns an integer containing one of three possible values: `JOptionPane.YES_OPTION`, `JOptionPane.NO_OPTION`, or `JOptionPane.CANCEL_OPTION`.

PTS: 1 REF: 89

11. Describe how the use of named constants can provide advantages over the use of literal values.

ANS:
Using named constants makes programs easier to read and understand.
When a constant is defined, you can change the constant at one location, which saves time and prevents you from missing other references.
Using named constants reduces typographical errors that may not be recognized by the compiler.
Constants can be easily identified when named conventionally (all caps).

PTS: 1 REF: 55

12. Describe why it is important to assign an appropriate data type to variables in an application.

ANS:
If a value is too large for the data type assigned, the compiler will issue an error message and the program will not execute.
If a data type is used that is larger than needed, memory is wasted.

PTS: 1 REF: 62-63

13. Describe how the `Scanner` class works with the `System.in` object in order to provide flexibility.

ANS:
The `System.in` object is designed to read bytes only. Since it is common to accept data of other types, the `Scanner` object can connect to the `System.in` property. This creates a `Scanner` object that will be connected to the default input device.

PTS: 1 REF: 76

14. `100 = salesAmount;`

In terms of assignment operators, why is the above statement illegal?

ANS:
This assignment operator has a right-to-left associativity. Associativity is the order in which values are used with operators. Since 100 is a numeric constant, it is an rvalue, which is an item that can appear only on the right side of the assignment operator. An identifier that can appear on the left side of an assignment operator is referred to as an lvalue (left-to-right associativity).

PTS: 1 REF: 53

15. Describe three ways in which a constant differs from a variable.

ANS:
Constants are preceded by the keyword `final` in a declaration statement.
Constants can be assigned a value once only and the value cannot be changed.
Constants conventionally have identifiers in all uppercase letters, distinguishing them from other forms.

PTS: 1 REF: 54

CASE

1. Write the statement to declare an uninitialized integer value for `salesAmt`.

   ANS:
   ```
   int salesAmt;
   ```

   The statement must end with a semicolon.

   PTS: 1 REF: 54

2. Write the statement that will declare and assign two integer variables, `salesAmt` and `costAmt`, in a single statement. Assign values of your choice to the variables.

   ANS:
   ```
   int salesAmt = 100, costAmt = 15;
   ```

   A semicolon must end the statement. Variable declarations are separated with a comma.

   PTS: 1 REF: 54

3.
   ```
   import javax.swing.JOptionPane;
   public class salesJune
   {
       public static void main(String[] args)
       {
          int storeSales = 250;
          ______________________________
       }
   }
   ```

   In the above code, complete the statement that will display a message dialog box that will appear centered on the screen and will display the following text:
   Congratulations! June sales were $250!

   ANS:
   ```
   JOptionPane.showMessageDialog(null, "Congratulations! June sales were
   $" + storeSales + "!");
   ```

   PTS: 1 REF: 57-58

4.
   ```
   final int COSTPERITEM = 10;
   double sales2012 = amtSold * COSTPERITEM;
   ```

In the above statements, identify the named constant and describe how a programmer can recognize named constants.

ANS:
The named constant identifier is `COSTPERITEM`.
Constant declaration statements use the `final` keyword.
Constants are conventionally given identifiers in all uppercase letters.

PTS: 1 REF: 54-55

5. Write the statement that will declare a `char` data type named `testScore` that will hold a letter grade of your choice.

ANS:
```
char testScore = ‘A’;
```

Letter assigned may vary.

PTS: 1 REF: 71

6.
```
public class YourGrade
{
    public static void main(String[] args)
    {
       int projectPoints = 89;
       System.out.print("Your grade for this class is ");
       System.out.print(projectPoints);
       System.out.println("%");
     }
}
```

Given the above code, what will be the output at the command prompt?

ANS:
Output will be as follows:

```
Your grade for this class is 89%
```

A blank line will follow the output.

PTS: 1 REF: 56-57

7. Describe the error message that will be produced when the following code is compiled.

```
public class SalesOct
{
    public static void main(String[] args)
    {
      int salesAmt;
      System.out.print("October sales are $");
      System.out.println(salesAmt);
     }
}
```

ANS:
The second `println` statement will generate an error message because the variable used in the statement is undeclared. It is legal to declare an uninitialized variable, but it cannot be used in a `println()` statement uninitialized. If you assign a numeric value to `int salesAmt`, the program will compile.

PTS: 1 REF: 61-62

8.
```
public class EndValue
{
    public static void main(String[] args)
    {
      int aByte = 940;
      System.out.print("The ending value is "+ aByte);
    }
}
```

When the above code is compiled, what error message will be generated and why?

ANS:
The above code will result in the error message “possible loss of precision”. The assigned value of 940 to the `aByte` variable is larger than the maximum value allowed. A `byte` type can hold a value between -128 and 127. Thus, the accuracy of the number has been compromised.

PTS: 1 REF: 62-65

9. Why is the following relational operator expression invalid? How could you rewrite the statement so that it is valid?

```
boolean isGradePassing = (grade => 70);
```

ANS:
In this statement, the order of the operator symbols is reversed. It is illegal to use =<, =>, and =!.

The statement could be modified as follows:

```
boolean isGradePassing = (grade >= 70);
```

PTS: 1 REF: 68

10.
```
char aCharacter = 2;
int aNumber = '2';
```

In the above statements, what values will be output after a `println()` statement is executed? Why are the output results different for the two statements?

ANS:
`aCharacter` will output a blank.
`aNumber` will output a value of 50.

Unicode values are used to assign a unique numeric code. Every computer stores each character it uses as a number and each character is assigned a unique Unicode numeric value.

PTS: 1 REF: 71

11. How could you alter the following statement to display “Welcome” on one line and “back” on another line?

```
System.out.println("Welcome back");
```

ANS:
There are two possible options:

```
System.out.println("Welcome\nback");
```

and

```
System.out.println("Welcome");
System.out.println("back");
```

PTS: 1 REF: 73-74