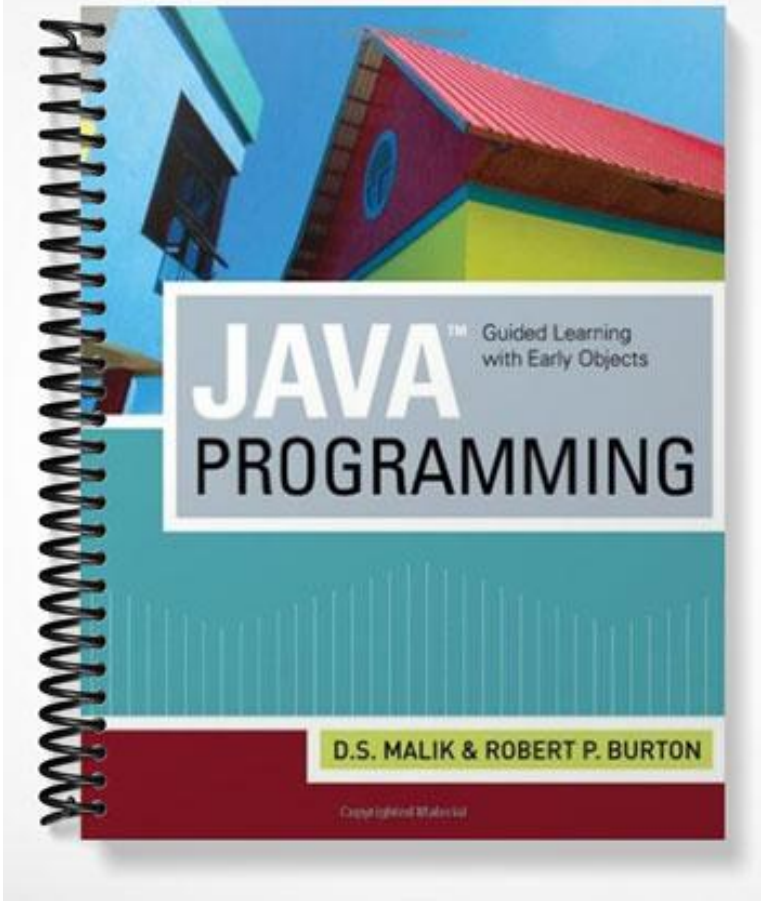
SOLUTIONS MANUAL



JAVA™ PROGRAMMING

Guided Learning with Early Objects

D.S. MALIK & ROBERT P. BURTON

Chapter 0

1. a. True; b. False; c. True; d. True; e. False; f; False; g. True; h. True; i. True;
2. Control unit and arithmetic logic unit.
3. Fetch and decode instructions, control the flow of information (instruction or data) between CPU and main memory, and control the operation of the internal components of the CPU.
4. Keyboard and mouse.
5. Monitor and printer.
6. Because programs and data must be stored in main memory before processing and because everything in main memory is lost when the computer is turned off, information stored in main memory must be transferred to some other device for permanent storage. Secondary storage stores programs and data long term.
7. Every computer directly understands its machine language. Therefore, for the computer to execute a program written in a high level language, the high level language program must be translated into the computer's machine language.
8. Instructions in a high level language are closer to the natural language, such as English, and therefore, are easier to understand and learn than the machine language.
9. A well-analyzed problem leads to a well designed algorithm. A well-designed algorithm can be coded directly. Moreover, a program that is well-analyzed is easier to modify and spot and fix errors.
10. To find the weighted average of the four test scores, first you need to know each test score and its weight. Next, you multiply each test score with its weight, and then add these numbers to find the `weightedAverage`. Therefore,
    1. Get `testScore1`, `weightTestScore1`
    2. Get `testScore2`, `weightTestScore2`
    3. Get `testScore3`, `weightTestScore3`
    4. Get `testScore4`, `weightTestScore4`
    5.
```
weightedAverage = testScore1 * weightTestScore1 +
                  testScore2 * weightTestScore2 +
                  testScore3 * weightTestScore3 +
                  testScore4 * weightTestScore4;
```
11. To find the average miles per gallon, first find the total distance traveled; this can be computed by subtracting starting odometer reading from the ending odometer reading. Next find the total amount of gasoline used by adding the amount gasoline used each day. To find average mile per gallon, divide the distance traveled by the amount of gasoline used. Therefore,
    1. Get `startOdometerReading`
    2. Get `endOdometerReading`
    3. `distanceTraveled = endOdometerReading- startOdometerReading`
    4. Get, `gUsedD1, gUsedD2, gUsedD3, gUsedD4, gUsedD5`
    5. `totalGasUsed = gUsedD1 + gUsedD2 + gUsedD3 + gUsedD4 + gUsedD5`
    6. `averageGasPerMile = distanceTraveled / totalGasUsed`

12. To calculate the selling price of the item, we need to know the price the store pays to buy the item. We can then use the following formula to find the selling price:

    ```
    sellingPrice = originalPrice + originalPrice * .60
    ```

    The algorithm is as follows:

    a. Get `originalPrice`

    b. Calculate the `sellingPrice` using the formula:

    ```
    sellingPrice = originalPrice + originalPrice * .60
    ```

    The information needed to calculate the selling price is the original price and the mark-up percentage.

13. To calculate the area of the triangle using the given formula, we need to know the lengths of the sides, `a`, `b`, and `c`, of the triangle. Next we calculate `s` using the formula

    ```
    s = (1/2)(a + b + c)
    ```

    next calculate the area using the formula:

    ```
    area = sqrt(s(s-a)(s-b)(s-c))
    ```

    where `sqrt` denote the square root.

    The algorithm is therefore:

    a. Get `a`, `b`, `c`

    b. `s = (1/2)(a + b + c)`

    c. `area = sqrt(s(s-a)(s-b)(s-c))`

    The information need to calculate the area of the triangle is the lengths of the sides of the triangle.

14. Let r denote the radius of the circle. Given the lengths a, b, and c, such that $a + b + c$ is the circumference of the circle, we have $2\pi r = a + b + c$. This implies that $r = (a + b + c) / (2\pi)$. We can now write the algorithm as follows:

    a. Get the values of `a`, `b`, `c`

    b. Calculate `r` using the formula:

    `r = (a + b + c) / (2π)`

15. Suppose `callTime` denotes the number of minutes the call lasted and `billingAmount` denotes the total charges for the call. To calculate the total charges for the call, you need to know the number of minutes the call lasted.

    If `callTime` is less than three minutes, the billing amount is connection charges plus $2.00; otherwise, billing amount is connection charges plus $2.00 plus 45 minutes per minutes over three minutes. That is,

    ```
    if (callTime is less than or equal to 3)
        billingAmount = 1.99 + 2.00;
    otherwise
        billingAmount = 1.99 + 2.00 + (callTime – 3)* 0.45;
    ```

    You can now write the algorithm as follows:

    a. Get `callTime`.

    b. Calculate billing amount using the formula:

    ```
    if (callTime is less than or equal to 3)
        billingAmount = 1.99 + 2.00;
    otherwise
    ```

```
billingAmount = 1.99 + 2.00 + (callTime – 3)* 0.45;
```

16. Suppose `averageTestScore` denotes the average test score, `highestScore` denotes the highest test score, `testScore` denotes a test score, `sum` denote the sum of all the test scores, and `count` denotes the number of students in class, and `studentName` denotes the name of a student.

    a. First you design an algorithm to find the average test score. To find the average test score, first you need to count the number of students in the class and add the test score of each student. You then divide the sum by count to find the average test score. The algorithm to find the average test score is as follows:

        i. Set `sum` and `count` to `0`.

        ii. Repeat the following for each student in class.

            1. Get `testScore`

            2. Increment `count` and update the value of `sum` by adding the current test score to `sum`.

        iii. Use the following formula to find the average test score.

```
if (count is 0)
    averageTestScore = 0;
otherwise
    averageTestScore = sum / count;
```

    b. The following algorithm determines and prints the names of all the students whose test score is below the average test score.

        Repeat the following for each student in class:

        i. Get `studentName` and `testScore`

        ii.

```
if (testScore is less than averageTestScore)
    print studentName
```

    c. The following algorithm determines and highest test score

        i. Get first student's test score and call it `highestTestScore`.

        ii. Repeat the following for each of the remaining student in class

            1. Get `testScore`

            2.

```
if (testScore is greater than highestTestScore)
    highestTestScore = testScore;
```

    d. To print the names of all the students whose test score is the same as the highest test score, compare the test score of each student with the highest test score and if they are equal print the name. The following algorithm accomplishes this

        Repeat the following for each student in class:

        i. Get `studentName` and `testScore`

        ii.

```
if (testScore is equal to highestTestScore)
    print studentName
```

You can use the solutions of the subproblems obtained in parts a to d to design the main algorithm as follows:

1. Use the algorithm in part (a) to find the average test score.
2. Use the algorithm in part (b) to print the names of all the students whose score is below the average test score.
3. Use the algorithm in part (c) to find the highest test score.
4. Use the algorithm in part (d) to print the names of all the students whose test score is the same as the highest test score

Chapter 1

1. a. False; b. False; c. False; d. False; e. True; f. True; g. False; h. True; i. False;
2. a, d, e, i, j
3. a
4. a. i; b. i; c. ii; d. iii; e. iv; f. i; g. iv;
5. a. `3;` b. `5.5;` c. `1;` d. `22.0;` e. `1;` f. `2;` g. `2;` h. `240.0;`
6. a, b, c, e, h, and j are valid;
   d, f, and g are invalid because the left side of an assignment must be a variable not an expression;
   i . `x % 5` evaluates to a decimal number while `n` is an `int` variable. The compiler will generate an error message indicating possible loss of precision.
   k. `3 + 4.6` evaluates to a decimal number while `n` is an `int` variable. The compiler will generate an error message indicating possible loss of precision.
7. `7`
8. Only the variable declarations in Lines 3 and 4 are correct The variable declaration in Line 1 is incorrect because the variable n has not been declared. The correct declaration is `int n = 12;` The variable declaration in Line 2 is incorrect because no character in single quotes precedes the semicolon. A correct declaration is `char letter = 'd';`
9. a and c are valid
10. a.
```
int x, y;
```
    b.
```
x = 10; ch = 'B';
```
    c.
```
x = x + 5;
```
    d.
```
z = 25.3;
```
    e.
```
z = y;
```
    f.
```
int temp = x;
x = y;
y = temp;
```
    g.
```
System.out.println(x + "  " + (2 * x + 5 - y));
```
    h.
```
char grade = 'A';
```
    i.
```
int num1, num2, num3, num4;
```
    j.
```
x = (int)(z + 0.5);
```
11. a.
```
10 * a
```
    b.
```
'8'
```
    c.
```
(b * b - 4 * a * c) / (2 * a)
```
    d.
```
(-b + (b * b - 4 * a * c)) / (2 * a)
```

12.
```
x = 5
y = 2
z = 3
w = 9
```

13.
```
x = 20
y = 15
z = 6
w = 11.5
t = 4.5
```

14.

	a	b	c
`a = (b++) + 3;`	9	7	UND
`c = 2 * a + (++b);`	9	8	26
`b = 2 * (++c) - (a++);`	10	45	27