Solutions to Quick Check Questions

6

Repetition Statements

6.1 The while Statement

1. Write a while statement to add numbers 11 through 20. Is this a count-controlled or sentinel-controlled loop?

   Answer:

```
int sum = 0, i = 11;

while ( i <= 20 ) { //this is a
    sum += i;        //count-controlled
    i++;
}
```

2. Write a while statement to read in real numbers and stop when a negative number is entered. Is this a count-controlled or sentinel-controlled loop?

```
Scanner scanner = new Scanner(System.in);
double num;

System.out.print("Enter number:");

num = scanner.nextDouble();
```

```
while ( num >= 0 ) { //this is a
                     //sentinel-controlled

   //do some operation using num

   System.out.print("Enter number:");

   num = scanner.nextDouble();
}
```

6.2 Pitfalls in Writing Repetition Statements

1. Which of the following is an infinite loop?

a.
```
int sum = 0, i = 0;
while ( i >= 0 ) {
   sum += i;
   i++;
}
```

Infinite Loop
i gets larger inside the loop.

b.
```
int sum = 0, i = 100;
while ( i != 0 ) {
   sum += i;
   i--;
}
```

Finite Loop
i gets smaller inside the loop and will become zero.

2. For each of the following loop statements, determine the value of sum after the loop is executed.

a.
```
int count = 0, sum = 0;
while ( count < 10 ) {
   sum += count;
   count++;
}
```

45

b.
```
int count = 1, sum = 0;
while ( count <= 30 ) {
   sum   += count;
   count += 3;
}
```

145

c.
```
int count = 0, sum = 0;
while ( count < 20 ) {
   sum   += 3*count;
   count += 2;
}
```

270

6.3 The do–while Statement

1. Write a do–while loop to compute the sum of the first 30 positive odd integers.

```
int num, sum = 0, i = 1;   /* A */

do {

   num = 2*i - 1; //get the i'th odd number

   sum += num;

   i++;

} while ( i <= 30);
```

```
int num = 1, sum = 0, i = 1;   /* B */

do {

   sum += num;

   num += 2;  //get the next odd number

   i++;

} while ( i <= 30);
```

2. Rewrite the following while loops as do–while loops.

a.
```
int count = 0, sum = 0;
while ( count < 10 ) {
   sum += count;
   count++;
}
```

Answer:

```
int count = 0, sum = 0;
do {
   sum += count;
   count++;
} while ( count < 10);
```

b.
```
int count = 1, sum = 0;
while ( count <= 30 ) {
    sum   += count;
    count += 3;
}
```

Answer:

```
int count = 1, sum = 0;
do {
    sum += count;
    count += 3;
} while ( count <= 30);
```

6.4 Loop-and-a-Half Repetition Control

1. Translate the following while loop to a loop-and-a-half format.

```
int sum = 0, num = 1;
while (num <= 50) {
    sum += num;
    num++;
}
```

Answer:

```
int sum = 0, num = 1;

while (true) {

    if (num > 50) break;

    sum += num;
    num++;
}
```

2. Translate the following do-while loop to a loop-and-a-half format.

```
int sum = 0, num = 1;
do {
    sum += num;
    num++;
} while (sum <= 5000);
```

Answer:

```
int sum = 0, num = 1;

while (true) {

    sum += num;
    num++;

    if (sum > 5000) break;

}
```

6.5 The for Statement

1. Write a for loop to compute

   a. the sum of 1, 2, . . . , 100.

   b. the sum of 2, 4, . . . , 500.

   c. the product of 5, 10, . . . , 50.

a.
```
sum = 0;
for (int i = 1; i <= 100; i++) {
   sum += i;
}
```

b.
```
sum = 0;
for (int i = 2; i <= 500; i+=2) {
   sum += i;
}
```

c.
```
sum = 0;
for (int i = 5; i <= 50; i+=5) {
   sum += i;
}
```

2. Rewrite the following while loops as for statements.

a.
```
int count = 0, sum = 0;
while ( count < 10 ) {
```

```
    sum += count;
    count++;
}
```

Answer:

```
sum = 0;
for (int count = 0; count < 10; count++) {
    sum += count;
}
```

b.

```
int count = 1, sum = 0;
while ( count <= 30 ) {
    sum   += count;
    count += 3;
}
```

Answer:

```
sum = 0;
for (int count = 1; count <= 30; count+=3) {
    sum += count;
}
```

6.6 Nested-for Statements

1. What will be the value of sum after the following nested-for loops are executed?

a.
```
int sum = 0;
for (int i = 0; i < 5; i++) {
    sum = sum + i;
    for (int j = 0; j < 5; j++) {
        sum = sum + j;
    }
}
```

60

b.
```
int sum = 0;
for (int i = 0; i < 5; i++) {
    sum = sum + i;
    for (int j = i; j < 5; j++) {
        sum = sum + j;
    }
}
```

50

2. What is wrong with the following nested-for loop?

```
int sum = 0;
for (int i = 0; i < 5; i++) {
   sum = sum + i;
   for (int i = 5; i > 0; i--) {
      sum = sum + j;
   }
}
```

The same variable i *is used in both loops. The variable* j *is not declared nor assigned an initial value.*

6.7 Formatting Output

1. Determine the output of the following code:

```
System.out.format("%3d + %3d = %3d", 1, 2, 3);
System.out.format("%tY", new Date());
System.out.format("%2$s,%1$s", "John", "Smith");
```

Answer:

```
1 +   2 =   32004Smith,John
```

Notice that the format method does not automatically moves to the next line. You need to add \n *at the end of the control string to do so.*

2. What's wrong with the following code?

```
Formatter f = new Formatter( );

f.format("%8.3f", 232.563);
```

The format method of the Formatter class returns a formatted string. It does not send the result to output. To send the result to the standard output, for example, we write

```
System.out.println(f.format("%8.3f", 232.563));
```

or

```
System.out.format("%8.3f", 232.563);
```

6.8 Loan Tables

No Quick Check Questions.

6.9 Estimating the Execution Time

No Quick Check Questions.

6.10 (Optional) Recursive Methods

No Quick Check Questions.

6.11 Sample Program: Hi-Lo Game

No Quick Check Questions.