Section 6

Chapter 4

Review Questions

Page 166- 171

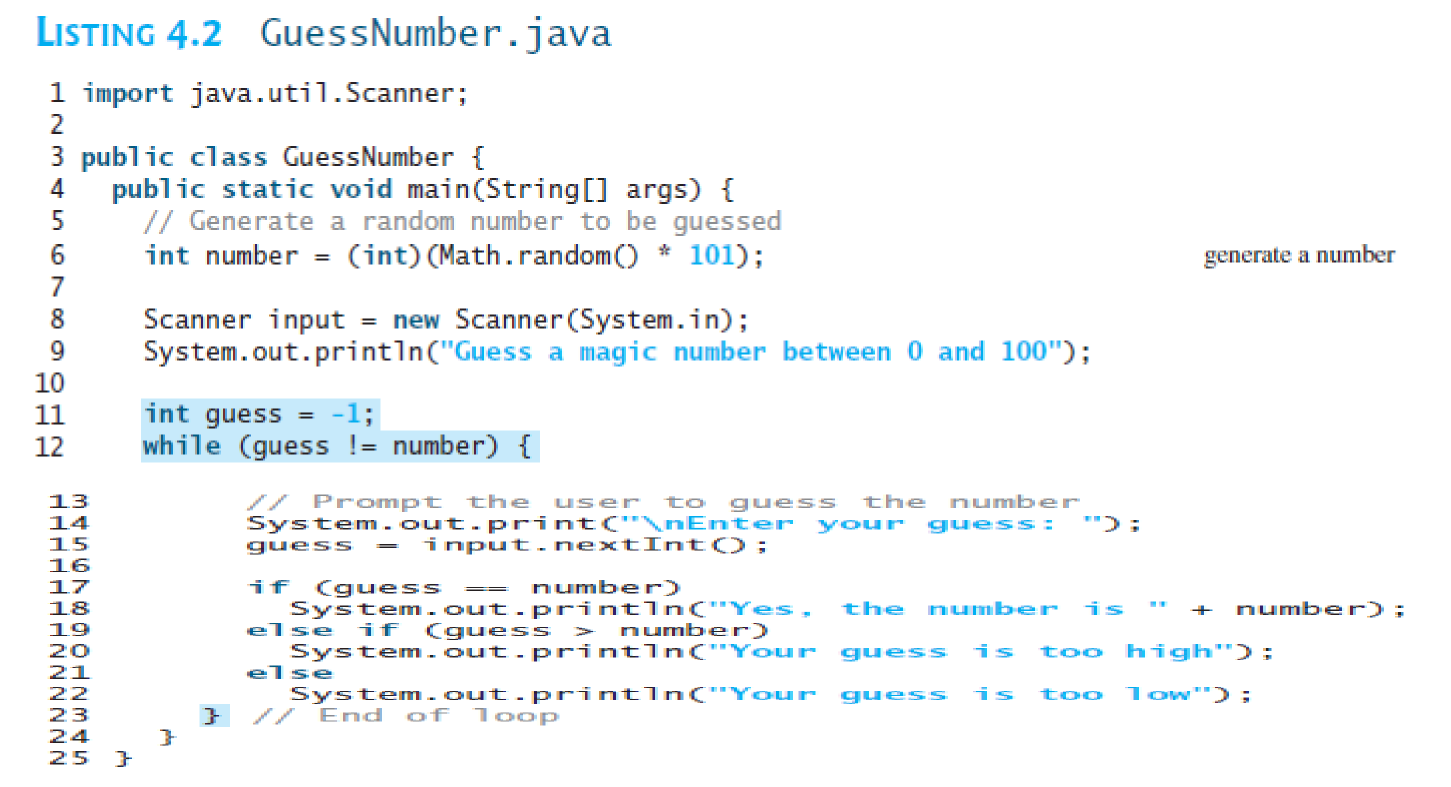
Review Questions 4.1

4.1 Analyze the following code. Is **count < 100 always true, always false, or** sometimes **true or sometimes false at Point A, Point B, and Point C?**

```
int count = 0;
while (count < 100) {
    // Point A  always true
  System.out.println("Welcome to Java!\n");
  count++;
  // Point B sometimes true or sometimes false
}
// Point C  Always false
```

Review Questions 4.2

4.2 What is wrong if **guess is initialized to 0 in line 11 in Listing 4.2?**



LISTING 4.2 GuessNumber.java

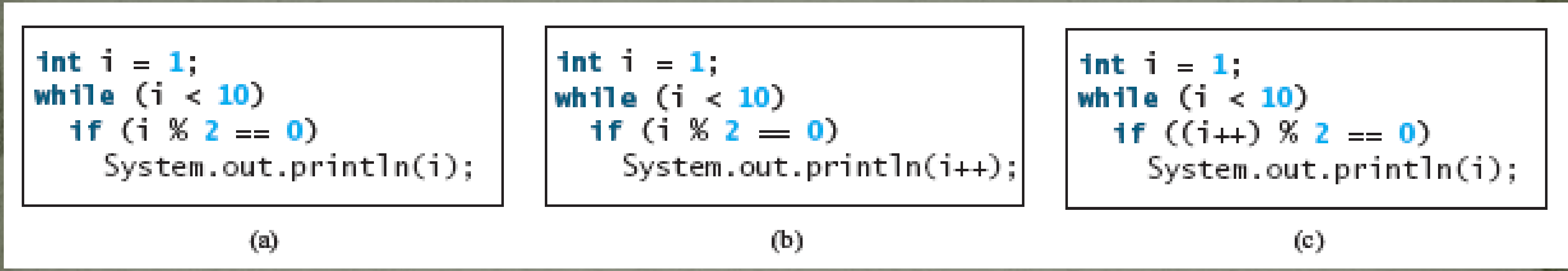
```
 1 import java.util.Scanner;
 2
 3 public class GuessNumber {
 4   public static void main(String[] args) {
 5     // Generate a random number to be guessed
 6     int number = (int)(Math.random() * 101);
 7
 8     Scanner input = new Scanner(System.in);
 9     System.out.println("Guess a magic number between 0 and 100");
10
11     int guess = -1;
12     while (guess != number) {
13       // Prompt the user to guess the number
14       System.out.print("\nEnter your guess: ");
15       guess = input.nextInt();
16
17       if (guess == number)
18         System.out.println("Yes, the number is " + number);
19       else if (guess > number)
20         System.out.println("Your guess is too high");
21       else
22         System.out.println("Your guess is too low");
23     } // End of loop
24   }
25 }
```

generate a number

It would be wrong if it is initialized to a value between 0 and 100, because it could be the number you attempt to guess.

Review Questions 4.3

4.3 How many times is the following loop body repeated? What is the printout of the loop?



```java
int i = 1;
while (i < 10)
  if (i % 2 == 0)
    System.out.println(i);
```

(a)

```java
int i = 1;
while (i < 10)
  if (i % 2 == 0)
    System.out.println(i++);
```

(b)

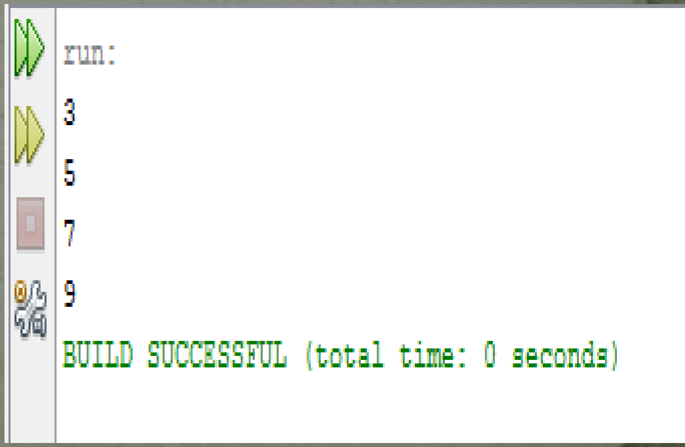
```java
int i = 1;
while (i < 10)
  if ((i++) % 2 == 0)
    System.out.println(i);
```

(c)

(a) Infinite number of times.

(b) Infinite number of times.

(c) nine times.



```
run:
3
5
7
9
BUILD SUCCESSFUL (total time: 0 seconds)
```

Review Questions 4.4.

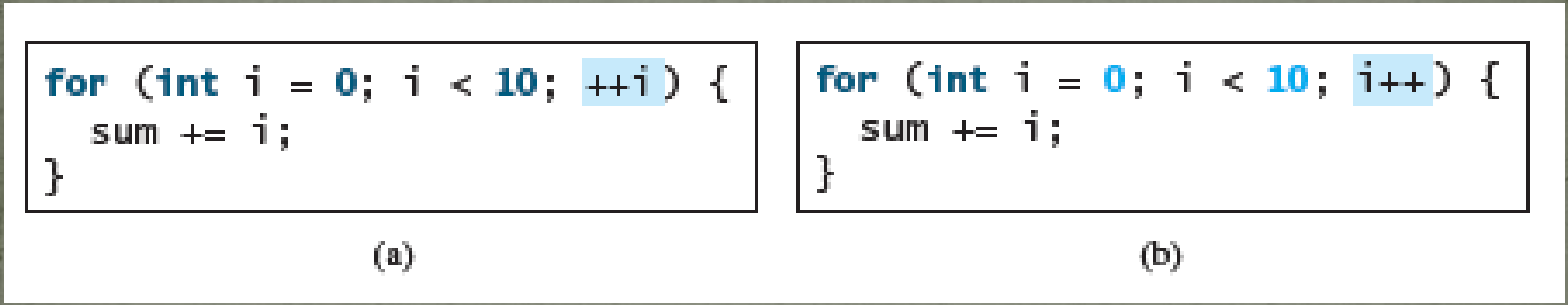
4.4 What are the differences between a **while loop and a do-while loop? Convert** the following **while loop into a do-while loop.**

```
int sum = 0;
int number = input.nextInt();
while (number != 0) {
    sum += number;
    number = input.nextInt();
}
```

```
int sum = 0;
int number;
do  {
    number = input.nextInt();
    sum += number;
}  while (number != 0) ;
```

Review Questions 4.5.

3.6 Do the following two loops result in the same value in sum?



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

(a)

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

(b)

Yes, the same value in sum.

Review Questions 4.7

4.7 Suppose the input is **2 3 4 5 0. W**
the following code?

```
1. Scanner input = new Scanner(System.in);
2. int number, max;
3. number = input.nextInt();
4. max = number;
5. while (number != 0) {
6.     number = input.nextInt();
 7.    if (number > max)
 8.        max = number;
9.  }
10.  System.out.println("max is " + max);
11.   System.out.println("number " + number)
```

step	number	max	output
3	2		
4		2	
6	3		
8		3	
6	4		
8		4	
6	5		
8		5	
6	0		
10			max is 5
11			number is 0

```
max is  5
number  0
```

Review Questio

4.8 Suppose the input is 2 3 4 5 0. Wh
the following code?

```
1. import java.util.Scanner;
2. publ
3. publ                                   g[] args) {
4. Scan                                   (System.in);
5.int number, sum = 0, count;
6. for (count = 0; count < 5; count++) {
7.    number = input.nextInt();
 8.   sum += number;
9. }
10. System.out.println("sum is " + sum
11. System.out.println("count is " + cou
} }
```

```
sum is  14
count is 5
```

step	number	sum	cont
5	0	0	
6			0
7	2		
8		2	
6			1
7	3		
8		5	
6			2
7	4		
8		9	
6			3
7	5		
8		14	
6			4
7	0		
8		14	
6			5

Review Questions 4.9

4.9 Suppose the input is **2 3 4 5 0**
the following code?

```
1. Scanner input = new Scanner(Syste
2. int number, max;
3. number = input.nextInt();
4. max = number;
5. do {
6.     number = input.nextInt();
7.     if (number > max)
8.         max = number;
9.  } while (number != 0)
10.  System.out.println("max is " + max);
11.   System.out.println("number " + number)
```

step	number	max	output
3	2		
4		2	
6	3		
8		3	
6	4		
8		4	
6	5		
8		5	
6	0		
10			max is 5
11			number is 0

```
max is 5
number 0
```

Review Questions 4.10

4.10 What does the following statement do?

```
for (   ;    ;   ) {
      do something;
}
```

The loop keeps doing something indefinitely

Review Questions 4.11

4.11 If a variable is declared in the **for loop control, can it be used after the loop exits?**

No. The scope of the variable is inside the loop.

Review Questions 4.12

4.12 Can you convert a **for loop to a while loop? List the advantages of using for** loops

Yes.
The advantages of for loops are simplicity and readability.
Compilers can produce more efficient code for the for loop than for the corresponding while loop.

Review Questions 4.13

4.13 Convert the following **for loop statement to a while loop and to a do-while loop:**

```
long sum = 0;
for (int i = 0; i <= 1000; i++)
sum = sum + i;
```

```
long sum = 0;
int i = 0;
while (i <= 1000)  {
sum = sum + i;
i++ ;
}
```

while loop

```
long sum = 0;
int i = 0;
do {
sum = sum + i;
i++ ;
} while (i <= 1000)
```

do-while

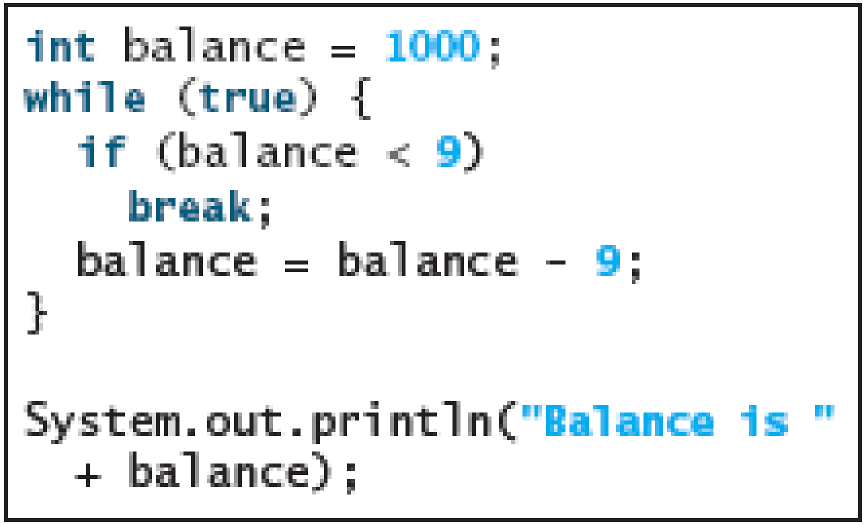
Review Questions 4.14

4.14 Will the program work if **n1 and n2 are replaced by n1 / 2 and n2 / 2 in line 17 in** Listing 4.8 (Greatest Common Divisor)?

No. Try n1 = 3 and n2 =3.

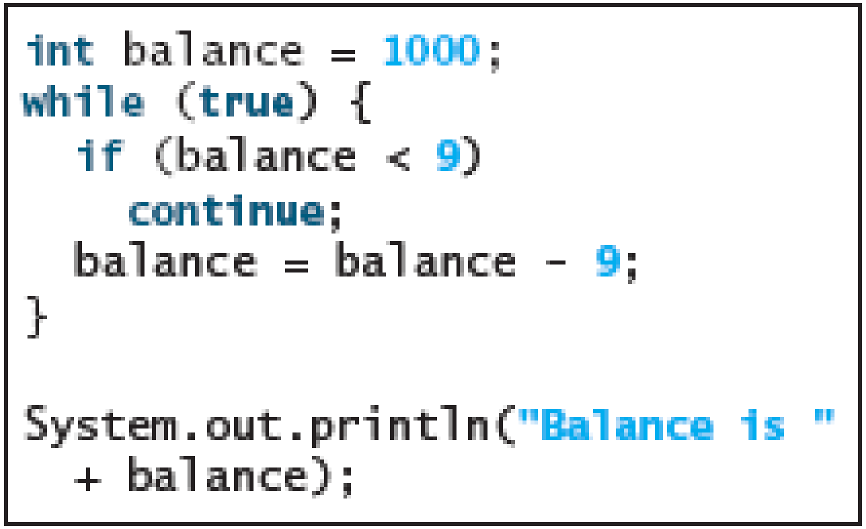
Review Questions 4.15

4.15 What is the keyword break for? What is the keyword continue for? Will the following program terminate? If so, give the output.



```
int balance = 1000;
while (true) {
  if (balance < 9)
    break;
  balance = balance - 9;
}

System.out.println("Balance is "
  + balance);
```

(a)



```
int balance = 1000;
while (true) {
  if (balance < 9)
    continue;
  balance = balance - 9;
}

System.out.println("Balance is "
  + balance);
```

(b)

(a) The program will terminate.

Balance is 1

(b) The program will not terminate

Review Questions 4.16

4.16 Can you always convert a **while loop into a for loop?** Yes

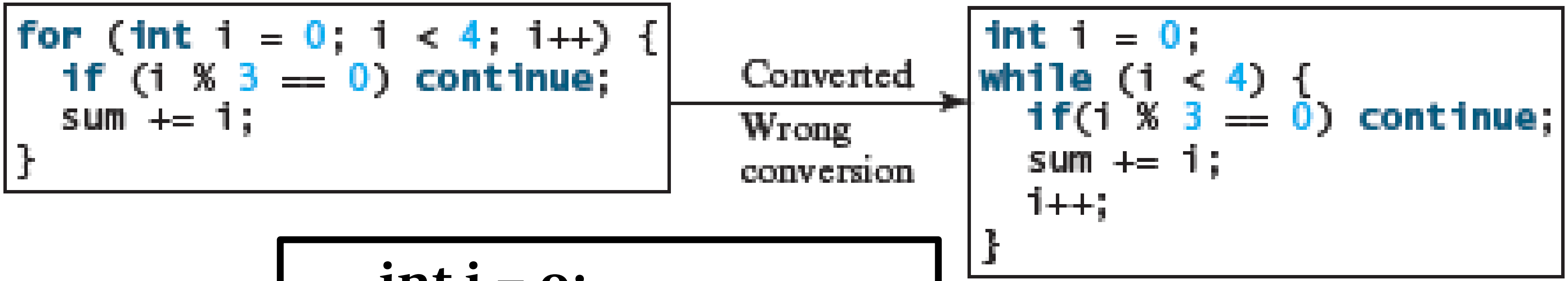
Convert the following while loop into a for loop.

```
int i = 1;
int sum = 0;
while (sum < 10000) {
sum = sum + i;
i++;
}
```

```
int sum = 0;
for (int i = 1; sum < 10000; i++)
      sum = sum + i;
```

Review Questions 4.17

4.17 The for loop on the left is converted into the while loop on the right. What is wrong? Correct it.



```
for (int i = 0; i < 4; i++) {
  if (i % 3 == 0) continue;
  sum += i;
}
```

```
int i = 0;
while (i < 4) {
  if(i % 3 == 0) continue;
  sum += i;
  i++;
}
```

```
int i = 0;
while (i < 4) {
  if (i % 3 == 0) {
     i++;
     continue;
  }
  sum += i;
  i++;
}
```

Review Questions 4.18

4.18 Rewrite the programs **TestBreak and TestContinue in Listings 4.11 and 4.12** without using **break and continue.**

```
public class TestBreak {
  public static void main(String[] args) {
    int sum = 0;
    int number = 0;

    while (number < 20) {
      number++;
      sum += number;
      if (sum >= 100)
        break;
    }

    System.out.println("The number is " + number);
    System.out.println("The sum is " + sum);
  }
}
```

```
while((number < 20) && (sum < 100)){
      number++;
      sum += number;
}
```

TestBreak

Review Questions 4.18 cont.

```
public class TestContinue {
  public static void main(String[] args) {
    int sum = 0;
    int number = 0;

    while (number < 20) {
      number++;
      if (number == 10 || number == 11)
          continue;
      sum += number;
    }

    System.out.println("The sum is " + sum);
  }
}
```

TestContinue

```
While (number < 20) {
         number++;
         if ((number != 10) && (number != 11))
            sum += number;
}
```

Review Questions 4.19

4.19 After the **break statement is exec** **following loop, which statement is** the output.

```
1.  for (int i = 1; i < 4; i++) {
2.    for (int j = 1; j < 4; j++) {
3.      if (i * j > 2)
4.          break;
5.      System.out.println(i * j);
6.  }
7. System.out.println(i);
8. }
```

```
1
2
1
2
2
3
```

step	i	j	output
1	1		
2		1	
5			1
2		2	
5			2
2	3		
7			1
1	2		
2		1	
5			2
7			2
1	3		
2		1	
5			3
1	4		

Review Question 4.

4.20 After the **continue statement is ex**
following loop, which statement is ex
the output.

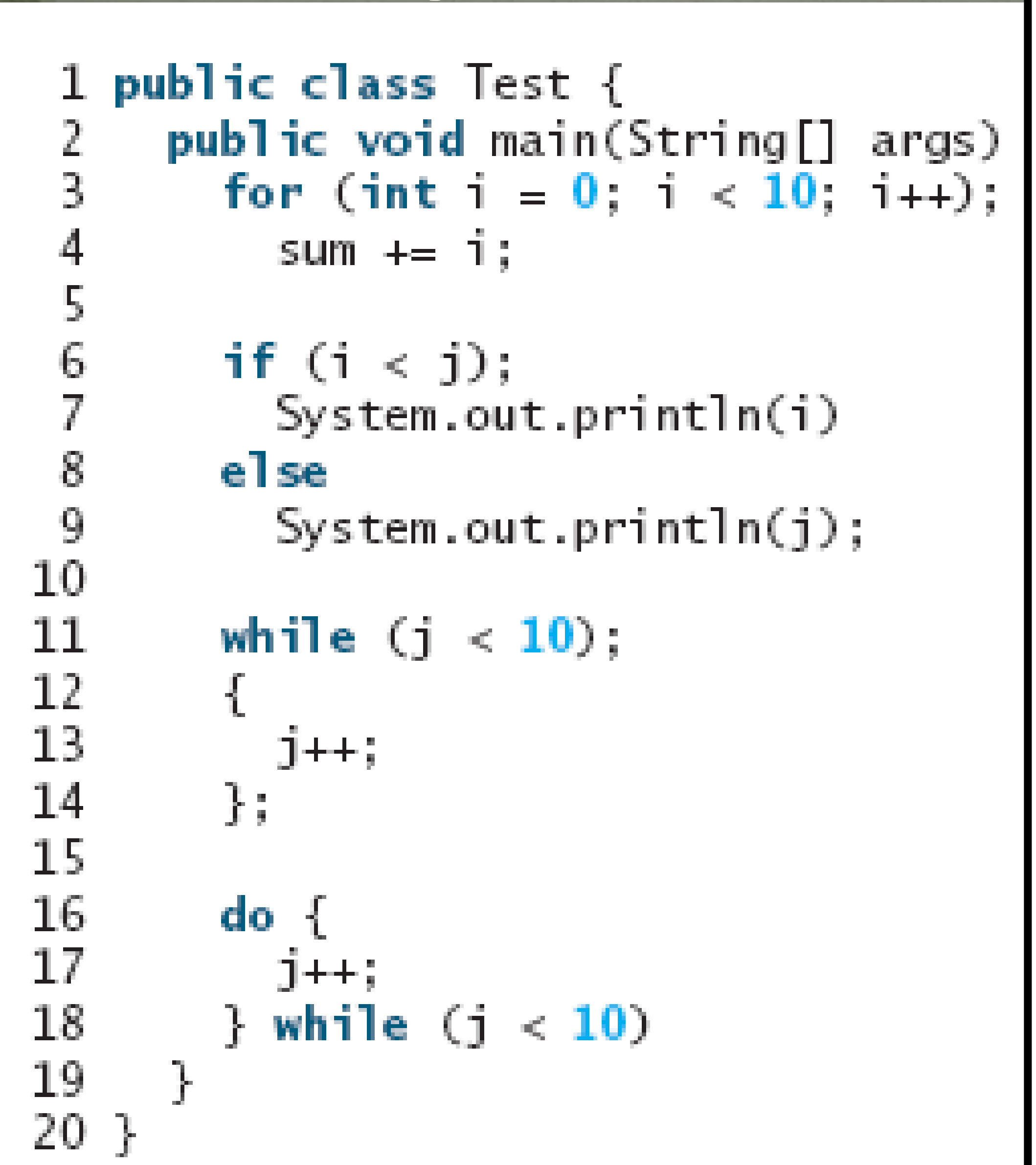
```
1.  for (int i = 1; i < 4; i++) {
2.     for (int j = 1; j < 4; j++) {
3.       if (i * j > 2)
4.          continue;
5.       System.out.println(i * j);
6.  }
7. System.out.println(i);
8. }
```

```
1
2
1
2
2
3
```

step	i	j	output
1	1		
2		1	
5			1
2		2	
5			2
2		3	
2		4	
7			1
1	2		
2		1	
5			2
2		2	
2		3	
2		4	
7			2
1	3		
2 :		1 :	
2 :		4 :	
7			3
1		4	

Review Ques

4.21 Identify and fix the errors i



```
 1 public class Test {
 2   public void main(String[] args)
 3     for (int i = 0; i < 10; i++);
 4       sum += i;
 5
 6     if (i < j);
 7       System.out.println(i)
 8     else
 9       System.out.println(j);
10
11     while (j < 10);
12     {
13       j++;
14     };
15
16     do {
17       j++;
18     } while (j < 10)
19   }
20 }
```

Line 2: missing static.

Line 3: The semicolon (;) at the end of the for loop heading should be removed.

Line 4: sum not defined.

Line 6: the semicolon (;) at the end of the if statement should be removed.

Line 6: j not defined.

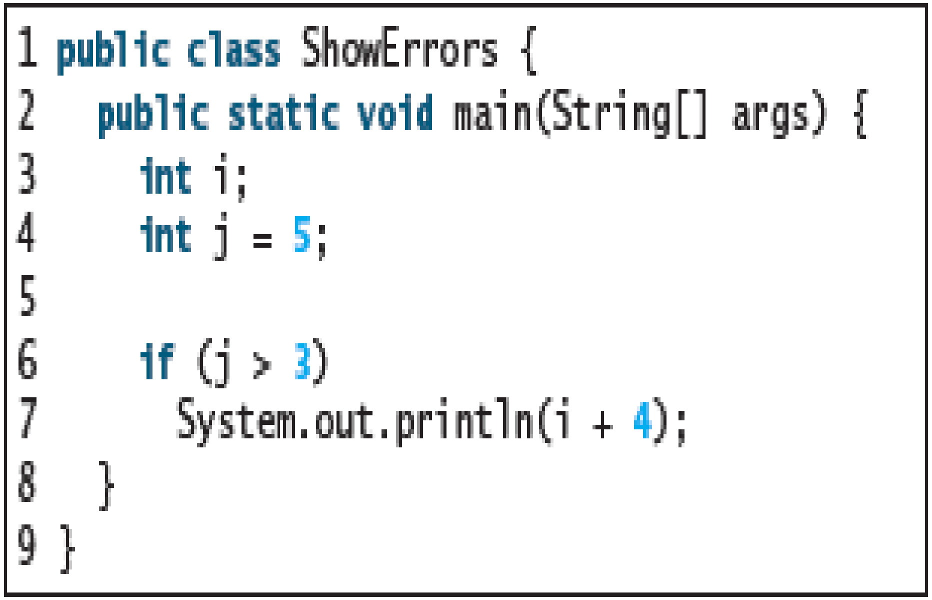
Line 7: Missing a semicolon for the first println statement.

Line 11: The semicolon (;) at the end of the while heading should be removed.

Line 18: Missing a semicolon at the end of the do-while loop.

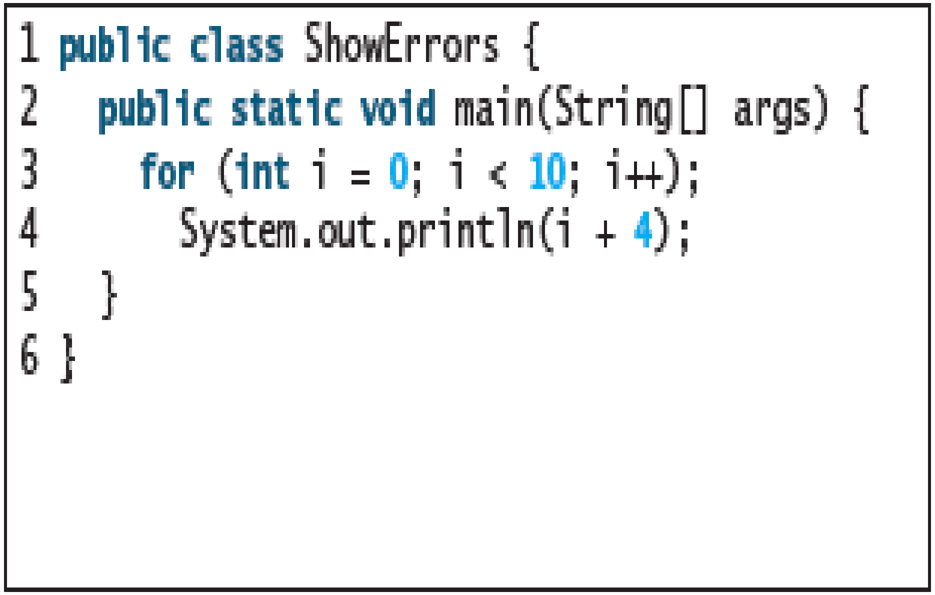
Review Questions 4.22

4.22 What is wrong with the following programs?



```
1 public class ShowErrors {
2   public static void main(String[] args) {
3     int i;
4     int j = 5;
5
6     if (j > 3)
7       System.out.println(i + 4);
8   }
9 }
```

(a)



```
1 public class ShowErrors {
2   public static void main(String[] args) {
3     for (int i = 0; i < 10; i++);
4       System.out.println(i + 4);
5   }
6 }
```

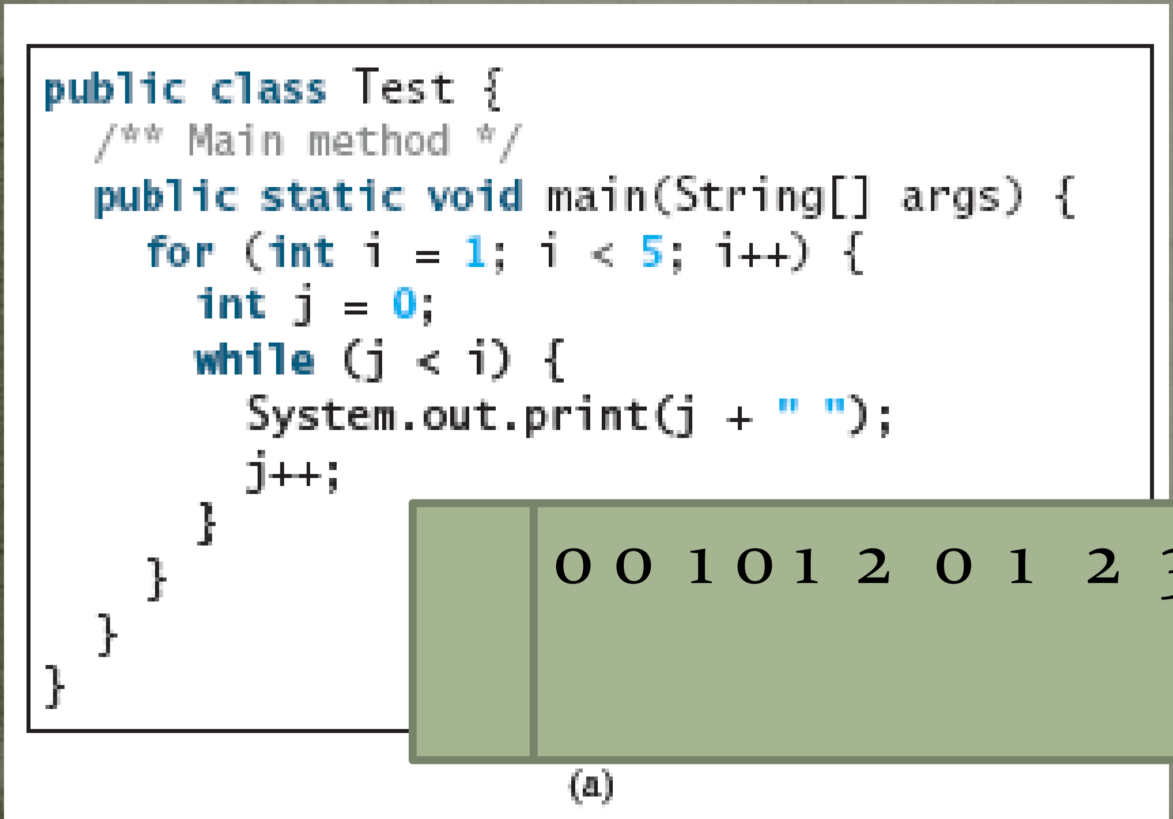
(b)

(a) Compile error (Syntax error): i is not initialized.

(b) Line 3: The ; at the end of for loop should be removed.

Review Questions 4.23

4.23 Show the output of the following prog ***(Ti*** ***Draw a table and list the variables*** in t trace these programs.)



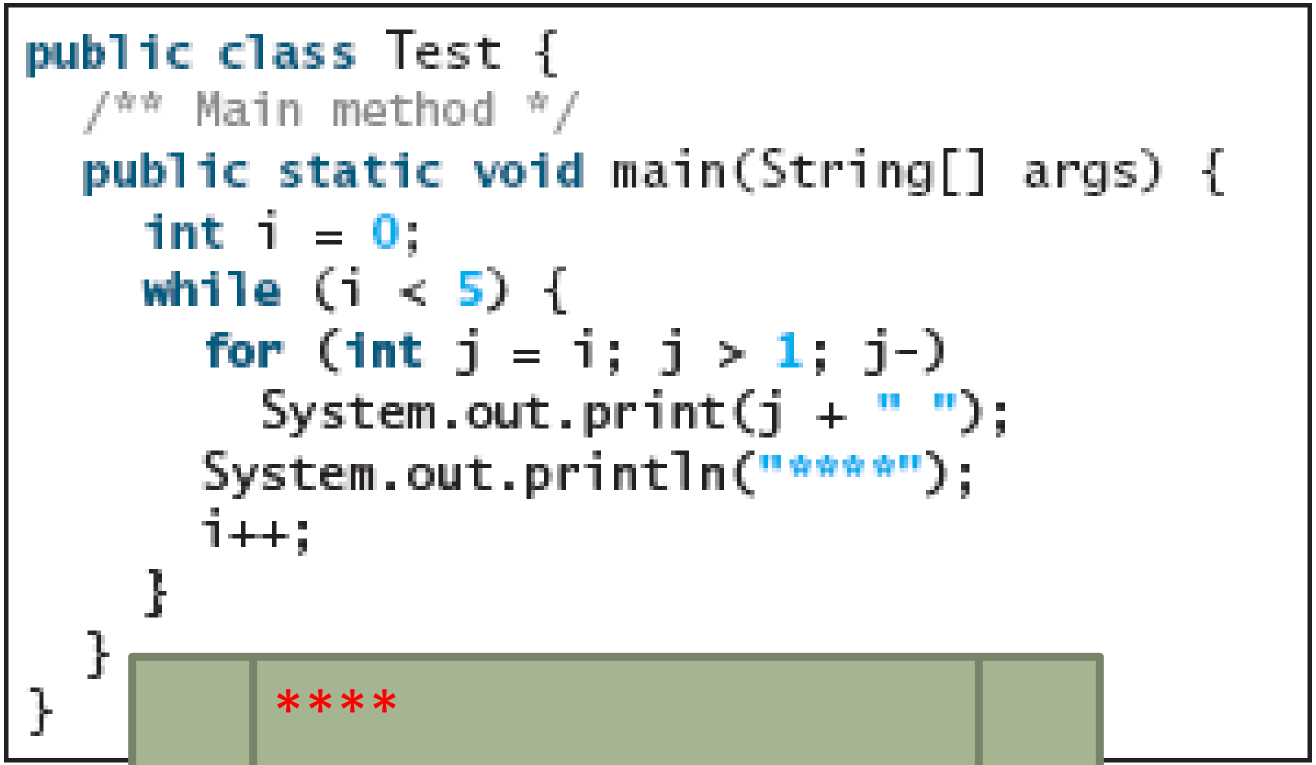
```
public class Test {
  /** Main method */
  public static void main(String[] args) {
    for (int i = 1; i < 5; i++) {
      int j = 0;
      while (j < i) {
        System.out.print(j + " ");
        j++;
      }
    }
  }
}
```

(a)

```
0 0 1 0 1 2 0 1 2 3
```

i	j	output
1	0	0
1	1	
2	0	0
2	1	1
2	2	
3	0	0
3	1	1
3	2	2
3	3	
4	0	0
4	1	1
4	2	2
4	3	3
4	4	
5		

Review Questions 4.23 cont.

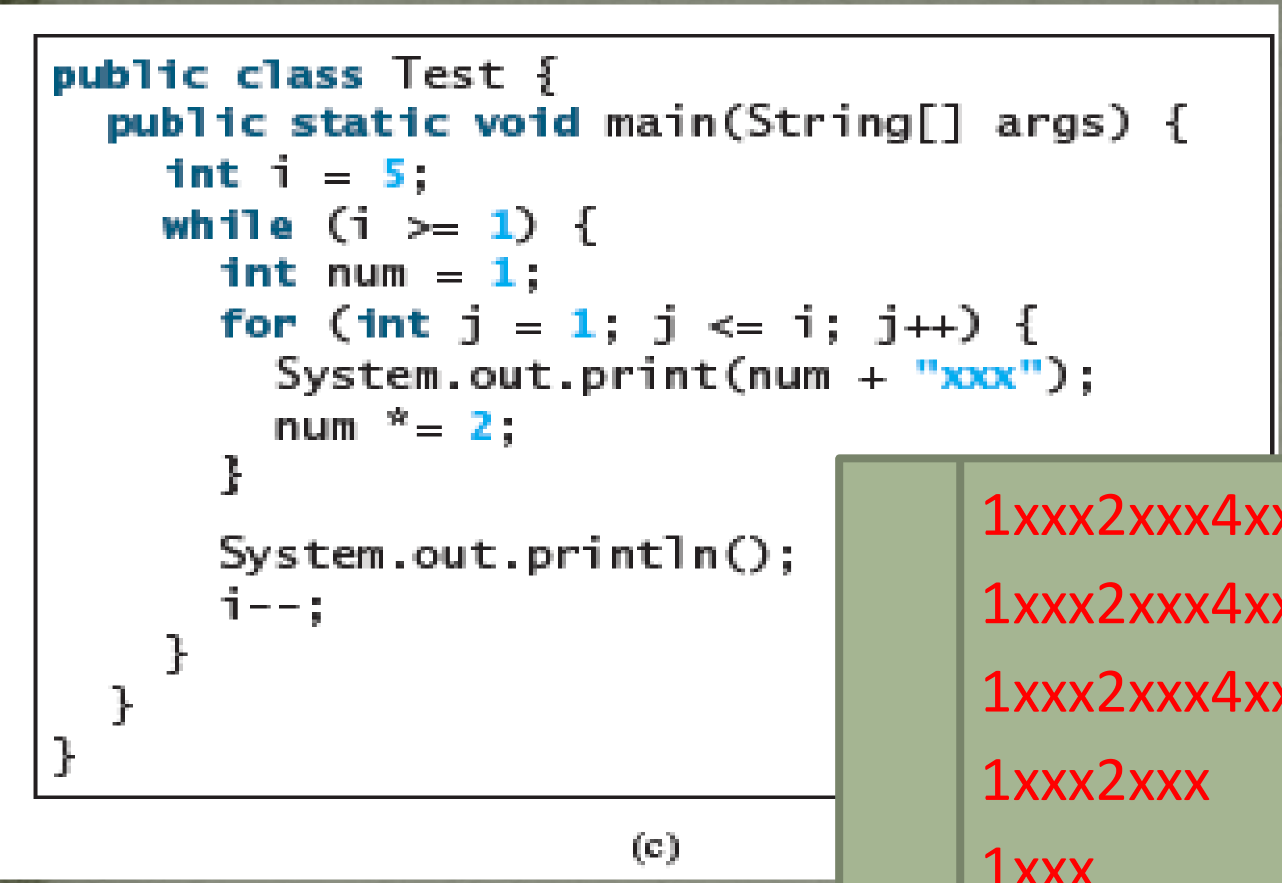


```
public class Test {
  /** Main method */
  public static void main(String[] args) {
    int i = 0;
    while (i < 5) {
      for (int j = i; j > 1; j-)
        System.out.print(j + " ");
      System.out.println("****");
      i++;
    }
  }
}
```

```
****

****

2 ****

3  2  ****

4  3  2  ****
```

i	j	output
0	0	****
1	1	****
2	2	2
2	1	****
3	3	3
3	2	2
3	1	****
4	4	4
4	3	3
4	2	2
4	1	****
5		

Review Questions 4.23 cont.

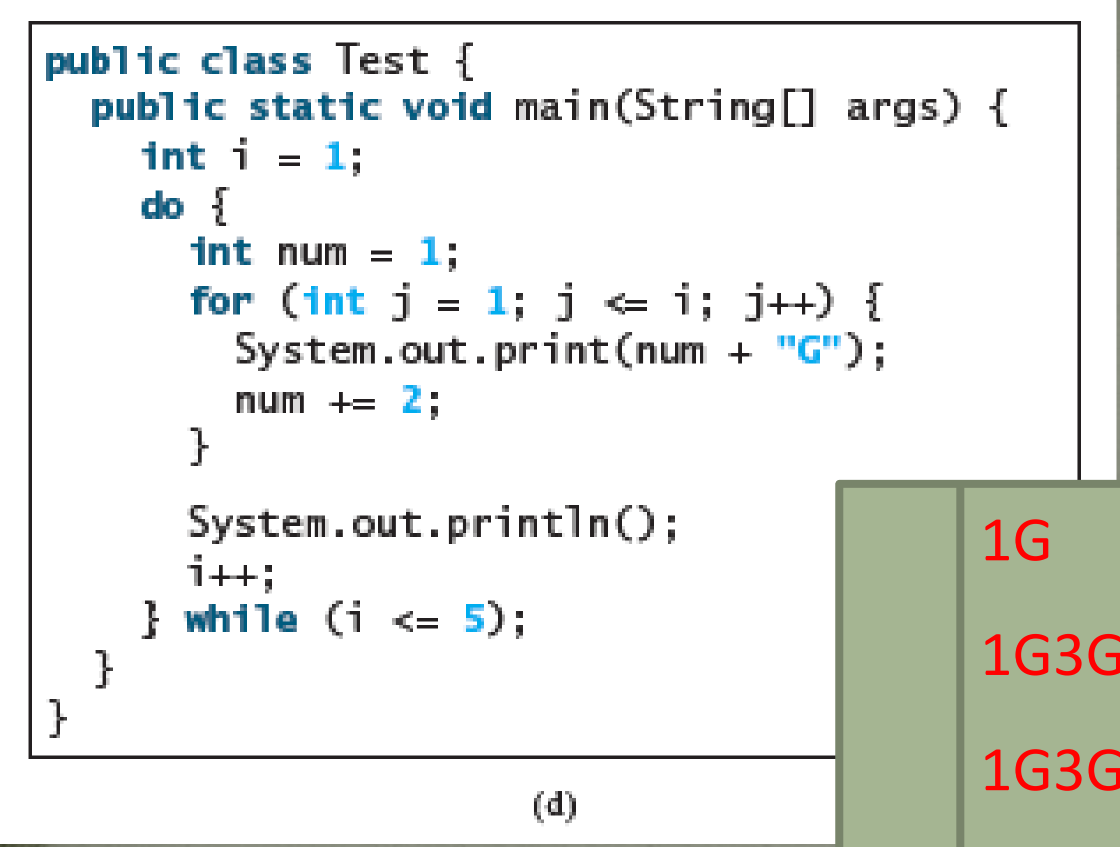


```
public class Test {
  public static void main(String[] args) {
    int i = 5;
    while (i >= 1) {
      int num = 1;
      for (int j = 1; j <= i; j++) {
        System.out.print(num + "xxx");
        num *= 2;
      }

      System.out.println();
      i--;
    }
  }
}
```

(c)

```
1xxx2xxx4xxx8xxx16xxx
1xxx2xxx4xxx8xxx
1xxx2xxx4xxx
1xxx2xxx
1xxx
```

Review Questions 4.23 cont.



```
public class Test {
  public static void main(String[] args) {
    int i = 1;
    do {
      int num = 1;
      for (int j = 1; j <= i; j++) {
        System.out.print(num + "G");
        num += 2;
      }

      System.out.println();
      i++;
    } while (i <= 5);
  }
}
```

(d)

```
1G
1G3G
1G3G5G
1G3G5G7G
1G3G5G7G9G
```

Review Questions 4.24

4.24 What is the output of the following program? Explain the reason.

```
int x = 80000000;
while (x > 0)
        x++;
System.out.println("x is " + x);
```

```
x is -2147483648
```

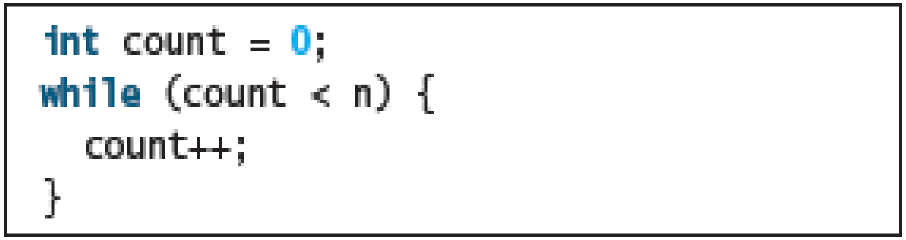
The reason:

When a variable is assigned a value that is too large *(in size)* to be stored, it causes *overflow*.

2147483647 + 1 is actually -2147483648

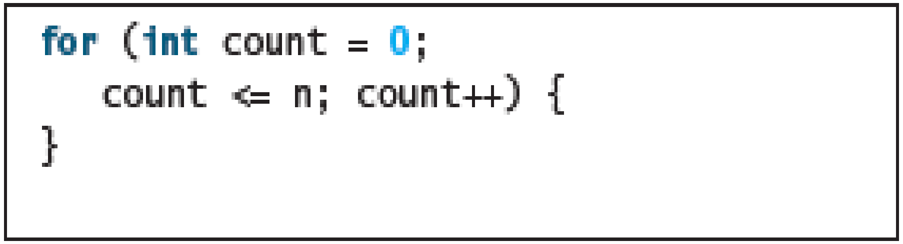
Review Questions 4.25

4.25 Count the number of iterations in the following loops.



```
int count = 0;
while (count < n) {
  count++;
}
```

(a)

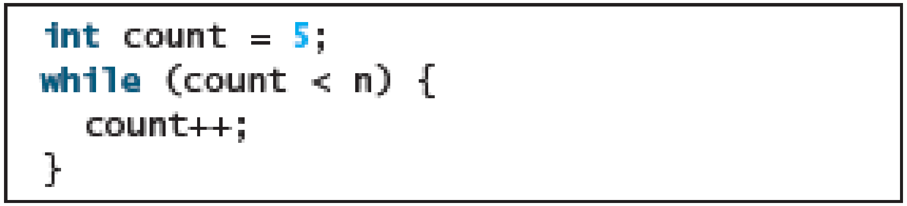


```
for (int count = 0;
   count <= n; count++) {
}
```

(b)

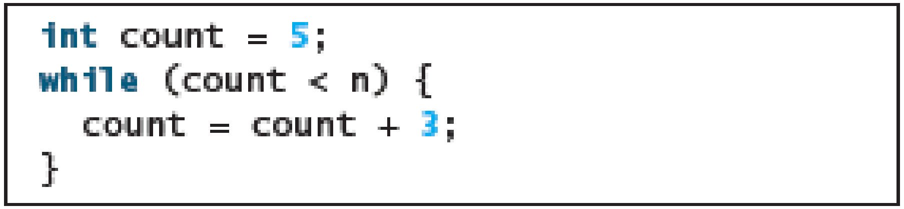
(a) n times

(b) n+1 times



```
int count = 5;
while (count < n) {
  count++;
}
```

(c)



```
int count = 5;
while (count < n) {
  count = count + 3;
}
```

(d)

(c) n-5 times

(c) (n-5)/3 times

Programming Exercises

Page 171-178