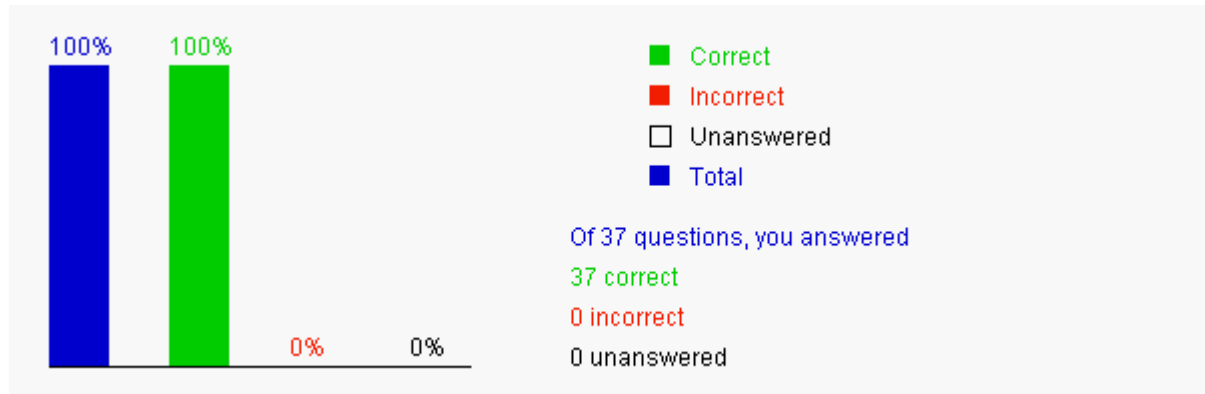


This quiz is for students to practice. A large number of additional quiz is available for instructors using Quiz Generator from the Instructor's Resource Website. Videos for Java, Python, and C++ can be found at <https://yongdanielliang.github.io/revelvideos.html>.

Chapter 5 Loops



Please send suggestions and errata to Dr. Liang at y.daniel.liang@gmail.com. Indicate which book and edition you are using. Thanks!

Section 5.2 The while Loop

5.1 How many times will the following code print "Welcome to Java"?

```
int count = 0;
while (count < 10) {
    System.out.println("Welcome to Java");
    count++;
}
```

- ☐ A. 8
- ☐ B. 9
- ☒ C. 10
- ☐ D. 11
- ☐ E. 0

Your answer is correct



Explanation: The count is initialized to 0 before the loop. The loop is executed 10 times for count from 1 to 9. When count is 10, the loop continuation condition becomes false. The loop is finished. So, the correct answer is C.

5.2 Analyze the following code.

```
int count = 0;
while (count < 100) {
    // Point A
    System.out.println("Welcome to Java!");
    count++;
    // Point B
}
```

// Point C

- ☒ A. count < 100 is always true at Point A
- ☐ B. count < 100 is always true at Point B
- ☐ C. count < 100 is always false at Point B
- ☐ D. count < 100 is always true at Point C
- ☒ E. count < 100 is always false at Point C

Your answer is correct



Explanation: The count is initialized to 0 before the loop. The loop is executed 100 times for count from 0 to 99. Inside the loop body, at Point A, count < 100 is true. After executing count++, count < 100 in Point B may become false. After the loop is exited, count < 100 is false at Point C. So, the correct answer is AE.

Section 5.3 Case Study: Guessing Numbers

5.3 How many times will the following code print "Welcome to Java"?

```
int count = 0;
while (count++ < 10) {
    System.out.println("Welcome to Java");
}
```

- ☐ A. 8
- ☐ B. 9

- ☒ C. 10
- ☐ D. 11
- ☐ E. 0



Your answer is correct

Explanation: The count is initialized to 0 before the loop. (count++ < 10) increments count by 1 and uses the old count value to check if count < 10. So, the loop is executed 10 times for count from 0 to 9. The correct answer is C.

5.4 What is the output of the following code?

```
int x = 0;
while (x < 4) {
    x = x + 1;
}
System.out.println("x is " + x);
```

- ☐ A. x is 0
- ☐ B. x is 1
- ☐ C. x is 2
- ☐ D. x is 3
- ☒ E. x is 4



Your answer is correct

Explanation: x is 0 before the loop. The loop is executed 4 times for x from 0 to 3. When x is 4 the loop exits. So, the correct answer is E.

Section 5.5 Controlling a Loop with User Confirmation or a Sentinel Value

5.5 What will be displayed when the following code is executed?

```
int number = 6;
while (number > 0) {
    number -= 3;
    System.out.print(number + " ");
}
```

- ☐ A. 6 3 0
- ☐ B. 6 3
- ☒ C. 3 0
- ☐ D. 3 0 -3
- ☐ E. 0 -3



Your answer is correct

Explanation: number is 6 before the loop. In the first iteration, number is reduced to 3. In the second iteration, number is reduced to 0. The loop is now finished. The loop body is executed 2 times for number 6 and 3. Since number is reduced by 3 before the print statement, 3 and 0 are displayed. So, the correct answer is C.

Section 5.6 The do-while Loop

5.6 How many times will the following code print "Welcome to Java"?

```
int count = 0;
do {
    System.out.println("Welcome to Java");
    count++;
} while (count < 10);
```

- ☐ A. 8
- ☐ B. 9
- ☒ C. 10
- ☐ D. 11
- ☐ E. 0



Your answer is correct

Explanation: The count is initialized to 0 before the loop. The loop is executed 10 times for count from 0 to 9. When count is 10, the loop continuation condition becomes false. The loop is finished. So, the correct answer is C.

5.7 How many times will the following code print "Welcome to Java"?

```
int count = 0;
do {
    System.out.println("Welcome to Java");
} while (count++ < 10);
```

- ☐ A. 8
- ☐ B. 9
- ☐ C. 10

- ☒ D. 11
☐ E. 0

Your answer is correct



Explanation: The count is initialized to 0 before the loop. The loop is executed for the first time when count is 0. (count++ < 10) increments count by 1 and uses the old count value to check if count < 10. So, the loop is executed 11 times for count from 0 to 10. Note that when count is 10, the loop body is executed before the loop is exited. The correct answer is D.

5.8 How many times will the following code print "Welcome to Java"?

```
int count = 0;
do {
    System.out.println("Welcome to Java");
} while (++count < 10);
```

- ☐ A. 8
☐ B. 9
☒ C. 10
☐ D. 11
☐ E. 0

Your answer is correct



Explanation: The count is initialized to 0 before the loop. The loop is executed for the first time when count is 0. (++count < 10) increments count by 1 and uses the new count value to check if count < 10. So, the loop is executed 10 times for count from 0 to 9. Note that when count is 9, ++count becomes 10, the loop exits. The correct answer is C.

5.9 What is the value in count after the following loop is executed?

```
int count = 0;
do {
    System.out.println("Welcome to Java");
} while (count++ < 9);
System.out.println(count);
```

- ☐ A. 8
☐ B. 9
☒ C. 10
☐ D. 11
☐ E. 0

Your answer is correct



Explanation: The count is initialized to 0 before the loop. The loop is executed for the first time when count is 0. (count++ < 9) increments count by 1 and uses the old count value to check if count < 9. So, the loop is executed 10 times for count from 0 to 9. Note that when count is 9, the loop body is executed and then count++ increments count to 10. The previous count value 9 < 9 is false. So the loop is finished. Now count becomes 10 after the loop exits. The correct answer is C.

Section 5.7 The for Loop

5.10 Analyze the following statement:

```
double sum = 0;
for (double d = 0; d < 10;) {
    d += 0.1;
    sum += sum + d;
}
```

- ☐ A. The program has a compile error because the adjustment is missing in the for loop.
☐ B. The program has a compile error because the control variable in the for loop cannot be of the double type.
☐ C. The program runs in an infinite loop because d < 10 would always be true.
☒ D. The program compiles and runs fine.

Your answer is correct



Explanation: In this loop, the loop initial action is d = 0, the continuation condition is d < 10, and the action-after-each-iteration is blank. Note that any of these three parts in the loop can be omitted. So, the loop is correct. The correction answer is (D).

5.11 Which of the following loops prints "Welcome to Java" 10 times?

A:

```
for (int count = 1; count <= 10; count++) {
    System.out.println("Welcome to Java");
}
```

B:

```
for (int count = 0; count < 10; count++) {
    System.out.println("Welcome to Java");
}
```

C:

```
for (int count = 1; count < 10; count++) {
    System.out.println("Welcome to Java");
}
```

D:

```
for (int count = 0; count <= 10; count++) {
    System.out.println("Welcome to Java");
}
```

- ☐ A. BD
☐ B. ABC
☐ C. AC
☐ D. BC
☒ E. AB

Your answer is correct 

Explanation: In (A), the loop displays Welcome to Java 10 times for count from 1 to 10. In (B), the loop displays Welcome to Java 10 times for count from 0 to 9. In (C), the loop displays Welcome to Java 9 times for count from 1 to 9. In (D), the loop displays Welcome to Java 11 times for count from 0 to 10. Therefore, the correct answer is AB.

5.12 Which of the following loops correctly computes $1/2 + 2/3 + 3/4 + \dots + 99/100$?

A:

```
double sum = 0;
for (int i = 1; i <= 99; i++) {
    sum = i / (i + 1);
}
System.out.println("Sum is " + sum);
```

B:

```
double sum = 0;
for (int i = 1; i < 99; i++) {
    sum += i / (i + 1);
}
System.out.println("Sum is " + sum);
```

C:

```
double sum = 0;
for (int i = 1; i <= 99; i++) {
    sum += 1.0 * i / (i + 1);
}
System.out.println("Sum is " + sum);
```

D:

```
double sum = 0;
for (int i = 1; i <= 99; i++) {
    sum += i / (i + 1.0);
}
System.out.println("Sum is " + sum);
```

E:

```
double sum = 0;
for (int i = 1; i < 99; i++) {
    sum += i / (i + 1.0);
}
System.out.println("Sum is " + sum);
```

- ☐ A. BCD
☐ B. ABCD
☐ C. B
☐ D. CDE
☒ E. CD

Your answer is correct 

Explanation: Note that $1/2$ is 0.5 in a math expression. In Java, however, integer division yields an integer. The fraction part is truncated. Therefore, $i/(i+1)$ is 0. (A) and (B) are incorrect. (E) is incorrect because the last i in the loop is 98. So the last item $99/100.0$ is not added to sum. So, the correct answer is CD.

5.13 The following loop displays _____.

```
for (int i = 1; i <= 10; i++) {
    System.out.print(i + " ");
    i++;
}
```

- ☐ A. 1 2 3 4 5 6 7 8 9
☐ B. 1 2 3 4 5 6 7 8 9 10
☐ C. 1 2 3 4 5
☒ D. 1 3 5 7 9

☐ E. 2 4 6 8 10

Your answer is correct



Explanation: In this for loop, i is initialized to 1. i++ adds 1 to i at the end of the iteration. In the action-after-each-iteration, i is again incremented by 1. So, i is in effect incremented twice. So, i is 1, then 3, 5, 7, and 9. The correct answer is D.

5.14 Do the following two statements in (I) and (II) result in the same value in sum?

(I):

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

(II):

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

- ☒ A. Yes
☐ B. No

Your answer is correct



Explanation: In this case, ++i and i++ are used standalone with no side-effect. So, (I) and (II) are equivalent. The correct answer is A.

5.15 What is the output for y?

```
int y = 0;  
for (int i = 0; i < 10; ++i) {  
    y += i;  
}  
System.out.println(y);
```

- ☐ A. 10
☐ B. 11
☐ C. 12
☐ D. 13
☒ E. 45

Your answer is correct



Explanation: The loop is executed 10 times for i from 0 to 9. Each time, i is added to y. So $y = 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$, which 45. The correct answer is E.

5.16 What is i after the following for loop?

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

- ☐ A. 9
☐ B. 10
☐ C. 11
☒ D. undefined

Your answer is correct



Explanation: The scope for i is inside the loop. After the loop, i is not defined. So, the correct answer is D.

5.17 Is the following loop correct?

```
for ( ; ; );
```

- ☒ A. Yes
☐ B. No

Your answer is correct



Explanation: Yes. This is equivalent to for (; true;).

Section 5.8 Which Loop to Use?

5.18 Analyze the following fragment:

```
double sum = 0;  
double d = 0;  
while (d != 10.0) {  
    d += 0.1;
```

```
sum += sum + d;
}
```

- ☐ A. The program does not compile because sum and d are declared double, but assigned with integer value 0.
- ☐ B. The program never stops because d is always 0.1 inside the loop.
- ☒ C. The program may not stop because of the phenomenon referred to as numerical inaccuracy for operating with floating-point numbers.
- ☐ D. After the loop, sum is $0 + 0.1 + 0.2 + 0.3 + \dots + 1.9$

Your answer is correct

Explanation: The correct answer is C. There is no guarantee that d will be exactly 10.0 because real numbers are represented using approximation in a computer system.

5.19 Analyze the following code:

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

- ☐ A. The program has a compile error because of the semicolon (;) on the for loop line.
- ☐ B. The program compiles despite the semicolon (;) on the for loop line, and displays 4.
- ☒ C. The program compiles despite the semicolon (;) on the for loop line, and displays 14.
- ☒ D. The for loop in this program is same as `for (i = 0; i < 10; i++) { }; System.out.println(i + 4);`

Your answer is correct

Explanation: This is a logic error. `System.out.println(i + 4)` is not a part for the for loop because the for loop ends with the last semicolon at `for(i=0; i < 10; i++);`

Section 5.9 Nested Loops

5.20 How many times is the println statement executed?

```
for (int i = 0; i < 10; i++)
    for (int j = 0; j < i; j++)
        System.out.println(i * j)
```

- ☐ A. 100
- ☐ B. 20
- ☐ C. 10
- ☒ D. 45

Your answer is correct

5.21 Which pattern is produced by the following code?

```
for (int i = 1; i <= 6; i++) {
    for (int j = 6; j >= 1; j--)
        System.out.print(j <= i ? j + " " : " " + " ");
    System.out.println();
}
```

Pattern A

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

Pattern B

```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Pattern C

```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
6 5 4 3 2 1
```

Pattern D

```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

- ☐ A. Pattern A
- ☐ B. Pattern B
- ☒ C. Pattern C
- ☐ D. Pattern D

Your answer is correct

Explanation: The outer loop displays a line. The inner loop displays all the numbers in the line. The outer loop is repeated 6 times for i from 1 to 6. For each i, the inner loops displays i as the first number in the line since j is from 6 to 1. j is printed for $j \leq i$. Otherwise, a space is printed. Clearly, this code will print Pattern C.

5.22 How many times is the println statement executed?

```
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        System.out.println(i * j);
```

- ☒ A. 100
- ☐ B. 20
- ☐ C. 10
- ☐ D. 45

Your answer is correct

Explanation: The outer loop is executed 10 times for i from 0 to 9 and the inner loop is executed 10 times from j from 0 to 9. So the print statement is executed 100 times. The correct answer is A.

Section 5.10 Minimizing Numerical Errors

5.23 To add $0.01 + 0.02 + \dots + 1.00$, what order should you use to add the numbers to get better accuracy?

- ☒ A. add 0.01, 0.02, ..., 1.00 in this order to a sum variable whose initial value is 0.
- ☐ B. add 1.00, 0.99, 0.98, ..., 0.02, 0.01 in this order to a sum variable whose initial value is 0.

Your answer is correct

Explanation: Adding the numbers in the order specified in A will result in a more accurate result than in B, because two numbers to be added will be about the same size.

5.24 Analyze the following code.

```
double sum = 0;
for (double d = 0; d < 10; sum += sum + d) {
    d += 0.1;
}
```

- ☐ A. The program has a syntax error because the adjustment statement is incorrect in the for loop.
- ☐ B. The program has a syntax error because the control variable in the for loop cannot be of the double type.
- ☐ C. The program compiles but does not stop because d would always be less than 10.
- ☒ D. The program compiles and runs fine.

Your answer is correct

Explanation: In this loop, the loop initial action is $d = 0$, the continuation condition is $d < 10$, and the action-after-each-iteration is $sum += sum + d$. The loop body is $d += 0.1$. This code is correct in syntax. d is initially 0 and $d += 0.1$ adds 0.1 to d in each iteration. Eventually d will be greater than or equal to 10. So the loop will terminate. The correct answer for this question is (D) The program compiles and runs fine.

Section 5.11 Case Studies

5.25 What is y after the following for loop statement is executed?

```
int y = 0;
for (int i = 0; i < 10; ++i) {
    y += 1;
}
```

- ☐ A. 9
- ☒ B. 10
- ☐ C. 11
- ☐ D. 12

Your answer is correct

Explanation: Before the loop, y is 0. The loop is executed 10 times. Each time, 1 is added to y. So, after the loop is finished, y is 10. The correct answer is (B).

Section 5.12 Keywords break and continue

5.26 Will the following program terminate?

```
int balance = 10;

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

- ☒ A. Yes
- ☐ B. No

Your answer is correct

Explanation: Yes. Before the loop, balance is 10. The loop-continuation-condition is always true. In the first iteration, balance is reduced to 1. In the second iteration, the break statement is executed since $(balance < 9)$ is now true. The correct answer for this question is A.

5.27 What is sum after the following loop terminates?

```
int sum = 0;
```

```

int item = 0;
do {
    item++;
    sum += item;
    if (sum > 4)
        break;
}
while (item < 5);

```

- ☐ A. 5
☒ B. 6
☐ C. 7
☐ D. 8
☐ E. 9

Your answer is correct



Explanation: sum and item are initialized to 0 before the loop. In the loop, item++ increments item by 1 and item is then added to sum. If sum > 4, the break statement exits the loop. item is initially 0, then 1, 2, 3 and sum is initially 0, and then 1, 3, and 6. When sum is 6, (sum > 4) is true, which causes the break statement to be executed. So, the correct answer is B.

5.28 What is the output after the following loop terminates?

```

int number = 25;
int i;

boolean isPrime = true;
for (i = 2; i < number && isPrime; i++) {
    if (number % i == 0) {
        isPrime = false;
    }
}

System.out.println("i is " + i + " isPrime is " + isPrime);

```

- ☐ A. i is 5 isPrime is true
☐ B. i is 5 isPrime is false
☐ C. i is 6 isPrime is true
☒ D. i is 6 isPrime is false

Your answer is correct



Explanation: The code tests if number is prime by dividing the number with possible divisors 2, 3, 4, and so on. Initially, isPrime is set to true. Once a divisor is found (i.e., number % i == 0 is true), isPrime is set to false. The loop continuation condition now becomes false. The loop ends. If no divisor is found after the loop is finished, isPrime remains true. Since number is 25, number % i == 0 is true when i is 5. In this case, isPrime is set to false and i++ increments i by 1 in the action-after-each-iteration. So, the output is that i is 6 and isPrime is false. The correct answer is D.

5.29 What is the output after the following loop terminates?

```

int number = 25;
int i;

boolean isPrime = true;
for (i = 2; i < number; i++) {
    if (number % i == 0) {
        isPrime = false;
        break;
    }
}

System.out.println("i is " + i + " isPrime is " + isPrime);

```

- ☐ A. i is 5 isPrime is true
☒ B. i is 5 isPrime is false
☐ C. i is 6 isPrime is true
☐ D. i is 6 isPrime is false

Your answer is correct



Explanation: The code tests if number is prime by dividing the number with possible divisors 2, 3, 4, and so on. Initially, isPrime is set to true. Once a divisor is found (i.e., number % i == 0 is true), isPrime is set to false and the break statement is then executed to exit the loop. If no divisor is found after the loop is finished, isPrime remains true. Since number is 25, number % i == 0 is true when i is 5. In this case, isPrime is set to false and the loop exits. So, the output is that i is 5 and isPrime is true. The correct answer is B.

5.30 What is sum after the following loop terminates?

```

int sum = 0;
int item = 0;
do {
    item++;
    if (sum >= 4)
        continue;
}
while (item < 5);

```



```

sum += item;
}
while (item < 5);

```

- ☒ A. 6
- ☐ B. 7
- ☐ C. 8
- ☐ D. 9
- ☐ E. 10

Your answer is correct



Explanation: In this loop, when $\text{sum} \geq 4$, the continue statement is executed to exit the current iteration so the next statement $\text{sum} += \text{item}$ after the if statement will not be executed. The loop keeps adding item to sum for item 1, 2, 3, and so on. So sum is 1, 2, and 6. When sum is 6, $\text{sum} \geq 4$ is true, the continue statement is executed to skip the rest of the iteration. When the loop ends, sum is still 6. So, the correct answer is A.

5.31 Will the following program terminate?

```

int balance = 10;

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

```

- ☐ A. Yes
- ☒ B. No

Your answer is correct



Explanation: Before the loop, balance is 10. In the loop, balance becomes 1. The loop-continuation-condition is always true. When $\text{balance} < 9$ is true, the continue statement is executed to skip the rest statement in the iteration. So the loop continues in an infinite loop. So, the correct answer is B.

5.32 What balance after the following code is executed?

```

int balance = 10;

while (balance >= 1) {
    if (balance < 9)
        continue;
    balance = balance - 9;
}

```

- ☐ A. -1
- ☐ B. 0
- ☐ C. 1
- ☐ D. 2
- ☒ E. The loop does not end

Your answer is correct



Explanation: Before the loop, balance is 10. The loop-continuation-condition is true ($10 \geq 1$). In the first iteration, balance is reduced to 1. Since $1 \geq 1$ is true, the loop body is executed. Since $\text{balance} < 9$ is true, the continue statement is executed to skip the rest statement in the iteration. balance continues to be 1 and the continue statement is executed to skip the rest statement in the iteration. So the loop runs infinitely. So the correct answer to this question is E.

5.33 What is the value of balance after the following code is executed?

```

int balance = 10;

while (balance >= 1) {
    if (balance < 9)
        break;
    balance = balance - 9;
}

```

- ☐ A. -1
- ☐ B. 0
- ☒ C. 1
- ☐ D. 2

Your answer is correct



Explanation: Before the loop, balance is 10. The loop-continuation-condition is true ($10 \geq 1$). In the first iteration, balance is reduced to 1. Since $1 \geq 1$ is true, the loop body is executed. Since $\text{balance} < 9$ is true, the break statement is executed to exit the loop. So, balance is 1 after the loop is finished. The correct answer for this question is C.

Section 5.13 Case Study: Checking Palindromes

5.34 What is the number of iterations in the following loop?

```
for (int i = 1; i < n; i++) {
    // iteration
}
```

☐ A. $2*n$
☐ B. n
☒ C. $n - 1$
☐ D. $n + 1$

Your answer is correct

Explanation: The loop is executed $n - 1$ times for i from 1 to $n - 1$. So, the correct answer is C.

5.35 What is the number of iterations in the following loop?

```
for (int i = 1; i <= n; i++) {
    // iteration
}
```

☐ A. $2*n$
☒ B. n
☐ C. $n - 1$
☐ D. $n + 1$

Your answer is correct

Explanation: The loop is executed n times for i from 1 to n . So, the correct answer is B.

Section 5.14 Case Study: Displaying Prime Numbers

5.36 Suppose the input for number is 9. What is the output from running the following program?

```
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int number = input.nextInt();

        int i;

        boolean isPrime = true;
        for (i = 2; i < number && isPrime; i++) {
            if (number % i == 0) {
                isPrime = false;
            }
        }

        System.out.println("i is " + i);

        if (isPrime)
            System.out.println(number + " is prime");
        else
            System.out.println(number + " is not prime");
    }
}
```

☐ A. i is 3 followed by 9 is prime
☐ B. i is 3 followed by 9 is not prime
☐ C. i is 4 followed by 9 is prime
☒ D. i is 4 followed by 9 is not prime

Your answer is correct

Explanation: The input number is 9. `isPrime` is initialized to true. The loop tests if number is divisible by i for i from 2, 3, and so on. When i is 3, `isPrime` is false. The loop continuation condition becomes false. The loop is not finished. Before the loop continuation condition is checked, `i++` increments i by 1. So, output is that i is 4 and following by 9 is not prime. So, the correct answer is D.

5.37 Analyze the following code:

```
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        int sum = 0;
        for (int i = 0; i < 100000; i++) {
            Scanner input = new Scanner(System.in);
            sum += input.nextInt();
        }
    }
}
```

- ☐ A. The program does not compile because the Scanner input = new Scanner(System.in); statement is inside the loop.
- ☐ B. The program compiles, but does not run because the Scanner input = new Scanner(System.in); statement is inside the loop.
- ☒ C. The program compiles and runs, but it is not efficient and unnecessary to execute the Scanner input = new Scanner(System.in); statement inside the loop. You should move the statement before the loop.
- ☐ D. The program compiles, but does not run because there is not prompting message for entering the input.

Your answer is correct



Explanation: To receive input from the keyboard, you need to create an input object from the Scanner class. You should create this object only once in the program. Placing the statement Scanner input = new Scanner(System.in) in the loop causes it to be created multiple times, which is a bad practice and could lead to potential errors. So, the correct answer is C.