U1-4_2

Name

1. Consider the following code segment.

```
int w = 1;
int x = w / 2;
double y = 3;
int z = (int) (x + y);
```

Which of the following best describes the results of compiling the code segment?

(A) The code segment compiles without error.

(B) The code segment does not compile, because the int variable x cannot be assigned the result of the operation w / 2.

(C) The code segment does not compile, because the integer value 3 cannot be assigned to the double variable y.

(D) The code segment does not compile, because the operands of the addition operator cannot be of different types int and double.

(E) The code segment does not compile because the result of the addition operation is of type double and cannot be cast to an int.

2. Consider the following code segment.

```
double x = 4.5;
int y = (int) x * 2;
System.out.print(y);
```

What is printed as a result of executing the code segment?

(A) 8

(B) 8.0

(C) 9

(D) 9.0

(E) 10

U1-4_2

3. Consider the following code segment, which is intended to simulate a random process. The code is intended to set the value of the variable event to exactly one of the values 1, 2, or 3, depending on the probability of an event occurring. The value of event should be set to 1 if the probability is 70 percent or less. The value of event should be set to 2 if the probability is greater than 70 percent but no more than 80 percent. The value of event should be set to 3 if the probability is greater than 80 percent. The variable randomNumber is used to simulate the probability of the event occurring.

```
int event = 0;
if (randomNumber <= 0.70)
{
event = 1;
}
if (randomNumber <= 0.80)
{
event = 2;
}
else
{
event = 3;
}
```

The code does not work as intended. Assume that the variable randomNumber has been properly declared and initialized. Which of the following initializations for randomNumber will demonstrate that the code segment will not work as intended?

(A) randomNumber = 0.70;

(B) randomNumber = 0.80;

(C) randomNumber = 0.85;

(D) randomNumber = 0.90;

(E) randomNumber = 1.00;

U1-4_2

4. Consider the following code segment.

```
int j = 1;
while (j < 5)
{
int k = 1;
while (k < 5)
{
System.out.println(k);
k++;
}
j++;
}
```

Which of the following best explains the effect, if any, of changing the first line of code to `int j = 0;` ?

(A) There will be one more value printed because the outer loop will iterate one additional time.

(B) There will be four more values printed because the outer loop will iterate one additional time.

(C) There will be one less value printed because the outer loop will iterate one fewer time.

(D) There will be four fewer values printed because the outer loop will iterate one fewer time.

(E) There will be no change to the output of the code segment.

U1-4_2

5. Consider the following method definition. The method printAllCharacters is intended to print out every character in str, starting with the character at index 0.

```
public static void printAllCharacters(String str)
{
for (int x = 0; x < str.length(); x++)   // Line 3
{
System.out.print(str.substring(x, x + 1));
}
}
```

The following statement is found in the same class as the printAllCharacters method.

```
printAllCharacters("ABCDEFG");
```

Which choice best describes the difference, if any, in the behavior of this statement that will result from changing x < str.length() to x <= str.length() in line 3 of the method?

Ⓐ The method call will print fewer characters than it did before the change because the loop will iterate fewer times.

Ⓑ The method call will print more characters than it did before the change because the loop will iterate more times.

Ⓒ The method call, which worked correctly before the change, will now cause a run-time error because it attempts to access a character at index 7 in a string whose last element is at index 6.

Ⓓ The method call, which worked correctly before the change, will now cause a run-time error because it attempts to access a character at index 8 in a string whose last element is at index 7.

Ⓔ The behavior of the code segment will remain unchanged.

6. The code segment below is intended to calculate the circumference c of a circle with the diameter d of 1.5. The circumference of a circle is equal to its diameter times pi.

```
/* missing declarations */
c = pi * d;
```

Which of the following variable declarations are most appropriate to replace /* *missing declarations* */ in this code segment?

(A)
```
int pi = 3.14159;
int d = 1.5;
final int c;
```

(B)
```
final int pi = 3.14159;
int d = 1.5;
int c;
```

(C)
```
final double pi = 3.14159;
double d = 1.5;
double c;
```

(D)
```
double pi = 3.14159;
double d = 1.5;
final double c = 0.0;
```

(E)
```
final double pi = 3.14159;
final double d = 1.5;
final double c = 0.0;
```

7. Consider the following code segment.

```
String temp = "comp";
System.out.print(temp.substring(0) + " " +
temp.substring(1) + " " +
temp.substring(2) + " " +
temp.substring(3));
```

What is printed when the code segment is executed?

(A) comp

(B) c o m p

(C) comp com co c

(D) comp omp mp p

(E) comp comp comp comp

U1-4_2

8. Consider the following Boolean expressions.

I.

A && B

II.

!A && !B

Which of the following best describes the relationship between values produced by expression I and expression II?

(A) Expression I and expression II evaluate to different values for all values of A and B.

(B) Expression I and expression II evaluate to the same value for all values of A and B.

(C) Expression I and expression II evaluate to the same value only when A and B are the same.

(D) Expression I and expression II evaluate to the same value only when A and B differ.

(E) Expression I and expression II evaluate to the same value whenever A is true.

9. Consider the following two code segments where the int variable choice has been properly declared and initialized.

Code Segment A

```
if (choice > 10)
{
System.out.println("blue");
}
else if (choice < 5)
{
System.out.println("red");
}
else
{
System.out.println("yellow");
}
```

Code Segment B

```
if (choice > 10)
{
System.out.println("blue");
}
if (choice < 5)
{
System.out.println("red");
}
else
{
System.out.println("yellow");
}
```

Assume that both code segments initialize choice **to the same integer value. Which of the following best describes the conditions on the initial value of the variable** choice **that will cause the two code segments to produce different output?**

U1-4_2

(A) choice < 5

(B) choice >= 5 and choice <= 10

(C) choice > 10

(D) choice == 5 or choice == 10

(E) There is no value for choice that will cause the two code segments to produce different output.

U1-4_2

10. Consider the following code segments, which are each intended to convert grades from a 100-point scale to a 4.0-point scale and print the result. A grade of 90 or above should yield a 4.0, a grade of 80 to 89 should yield a 3.0, a grade of 70 to 79 should yield a 2.0, and any grade lower than 70 should yield a 0.0.

Assume that grade is an int variable that has been properly declared and initialized.

Code Segment I

```
double points = 0.0;
if (grade > 89)
{
points += 4.0;
}
else if (grade > 79)
{
points += 3.0;
}
else if (grade > 69)
{
points += 2.0;
}
else
{
points += 0.0;
}
System.out.println(points);
```

Code Segment II

```
double points = 0.0;
if (grade > 89)
{
points += 4.0;
}
if (grade > 79)
{
grade += 3.0;
}
if (grade > 69)
{
points += 2.0;
}
if (grade < 70)
{
points += 0.0;
}
System.out.println(points);
```

Which of the following statements correctly compares the values printed by the two methods?

(A) The two code segments print the same value only when grade is below 80.

(B) The two code segments print the same value only when grade is 90 or above or grade is below 80.

(C) The two code segments print the same value only when grade is 90 or above.

(D) Both code segments print the same value for all possible values of grade.

(E) The two code segments print different values for all possible values of grade.

---

11. Consider the following code segments. Code segment 2 is a revision of code segment 1 in which the loop increment has been changed.

Code Segment 1

```
int sum = 0;
for (int k = 1; k <= 30; k++)
{
sum += k;
}
System.out.println("The sum is: " + sum);
```

Code Segment 2

```
int sum = 0;
for (int k = 1; k <= 30; k = k + 2)
{
sum += k;
}
System.out.println("The sum is: " + sum);
```

Code segment 1 prints the sum of the integers from 1 through 30, inclusive. Which of the following best explains how the output changes from code segment 1 to code segment 2 ?

(A) Code segment 1 and code segment 2 will produce the same output.

(B) Code segment 2 will print the sum of only the even integers from 1 through 30, inclusive because it starts sum at zero, increments k by twos, and terminates when k exceeds 30.

(C) Code segment 2 will print the sum of only the odd integers from 1 through 30, inclusive because it starts k at one, increments k by twos, and terminates when k exceeds 30.

(D) Code segment 2 will print the sum of only the even integers from 1 through 60, inclusive because it starts sum at zero, increments k by twos, and iterates 30 times.

(E) Code segment 2 will print the sum of only the odd integers from 1 through 60, inclusive because it starts k at one, increments k by twos, and iterates 30 times.

12. Consider the following code segment.

```
int a = 5;
int b = 4;
int c = 2;
a *= 3;
b += a;
b /= c;
System.out.print(b);
```

What is printed when the code segment is executed?

(A) 2

(B) 4

(C) 9

(D) 9.5

(E) 19

U1-4_2

13. Consider the following code segment.

```
String str = "CompSci";
System.out.println(str.substring(0, 3));
int num = str.length();
```

What is the value of num when the code segment is executed?

Ⓐ 3

Ⓑ 4

Ⓒ 5

Ⓓ 6

Ⓔ 7

14. Consider the following code segment.

```
String str = "0";
str += str + 0 + 8;
System.out.println(str);
```

What is printed as a result of executing the code segment?

Ⓐ 8

Ⓑ 08

Ⓒ 008

Ⓓ 0008

Ⓔ Nothing is printed, because numerical values cannot be added to a String object.

U1-4_2

15. Consider the following code segment.

```
int one = 1;
int two = 2;
String zee = "Z";
System.out.println(one + two + zee);
```

What is printed as a result of executing the code segment?

(A) 12Z

(B) 3Z

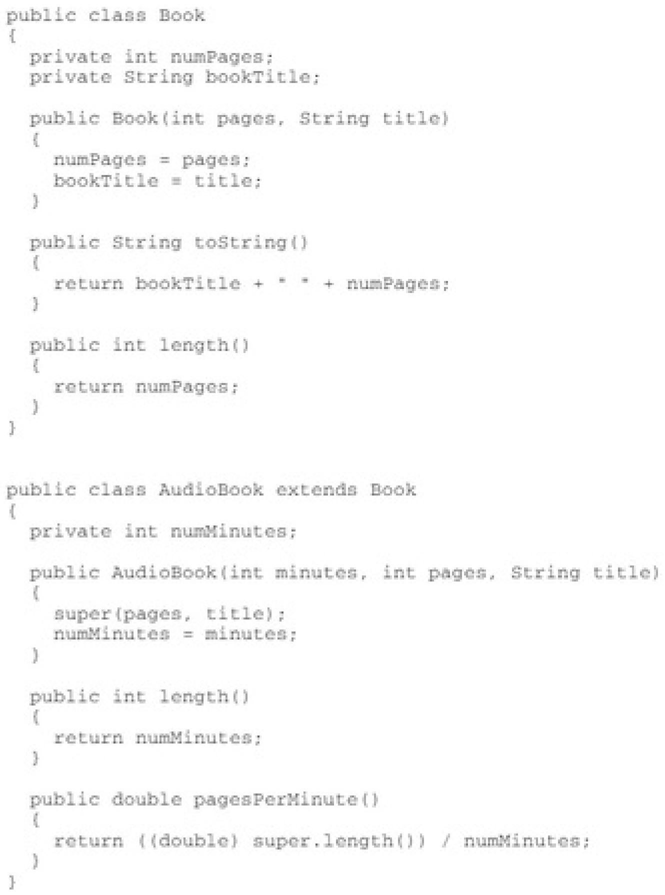
(C) 12zee

(D) 3zee

(E) onetwozee

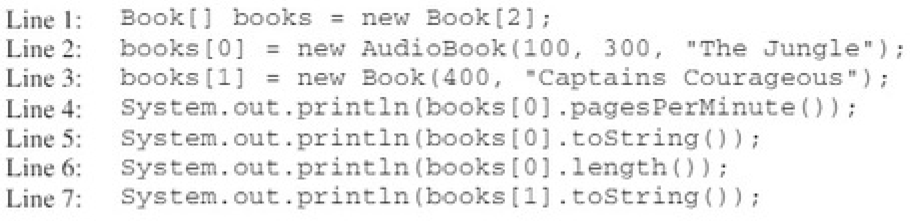
U1-4_2

16. Consider the following Book and AudioBook classes.



```
public class Book
{
  private int numPages;
  private String bookTitle;

  public Book(int pages, String title)
  {
    numPages = pages;
    bookTitle = title;
  }

  public String toString()
  {
    return bookTitle + " " + numPages;
  }

  public int length()
  {
    return numPages;
  }
}


public class AudioBook extends Book
{
  private int numMinutes;

  public AudioBook(int minutes, int pages, String title)
  {
    super(pages, title);
    numMinutes = minutes;
  }

  public int length()
  {
    return numMinutes;
  }

  public double pagesPerMinute()
  {
    return ((double) super.length()) / numMinutes;
  }
}
```

Consider the following code segment that appears in a class other than Book or AudioBook.



```
Line 1:   Book[] books = new Book[2];
Line 2:   books[0] = new AudioBook(100, 300, "The Jungle");
Line 3:   books[1] = new Book(400, "Captains Courageous");
Line 4:   System.out.println(books[0].pagesPerMinute());
Line 5:   System.out.println(books[0].toString());
Line 6:   System.out.println(books[0].length());
Line 7:   System.out.println(books[1].toString());
```

Which of the following best explains why the code segment will not compile?

A Line 2 will not compile because variables of type Book may not refer to variables of type AudioBook.

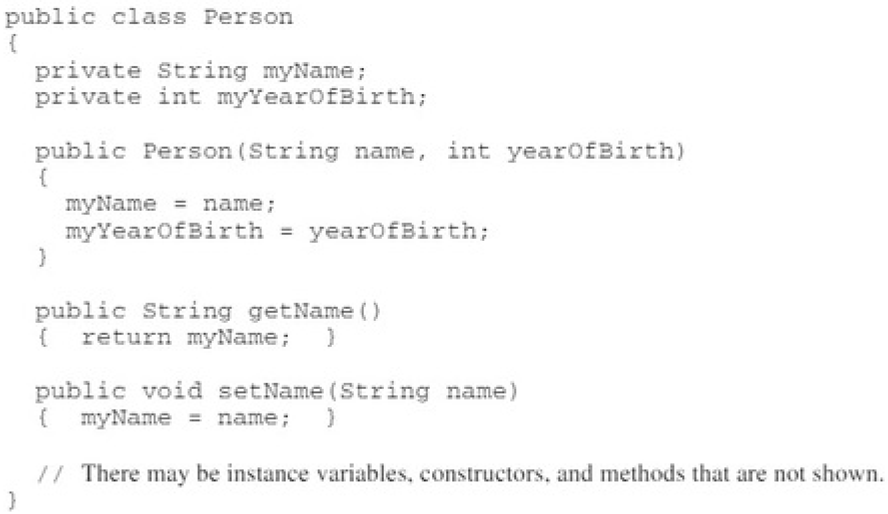
B Line 4 will not compile because variables of type Book may only call methods in the Book class.

C Line 5 will not compile because the AudioBook class does not have a method named toString declared or implemented.

D Line 6 will not compile because the statement is ambiguous. The compiler cannot determine which length method should be called.

E Line 7 will not compile because the element at index 1 in the array named books may not have been initialized.

17. Consider the following class declaration.



```
public class Person
{
  private String myName;
  private int myYearOfBirth;

  public Person(String name, int yearOfBirth)
  {
    myName = name;
    myYearOfBirth = yearOfBirth;
  }

  public String getName()
  {  return myName;  }

  public void setName(String name)
  {  myName = name;  }

  // There may be instance variables, constructors, and methods that are not shown.
}
```

Assume that the following declaration has been made.
Person student = new Person ("Thomas", 1995);
Which of the following statements is the most appropriate for changing the name of student from "Thomas" to "Tom" ?

Ⓐ student = new Person ("Tom", 1995);

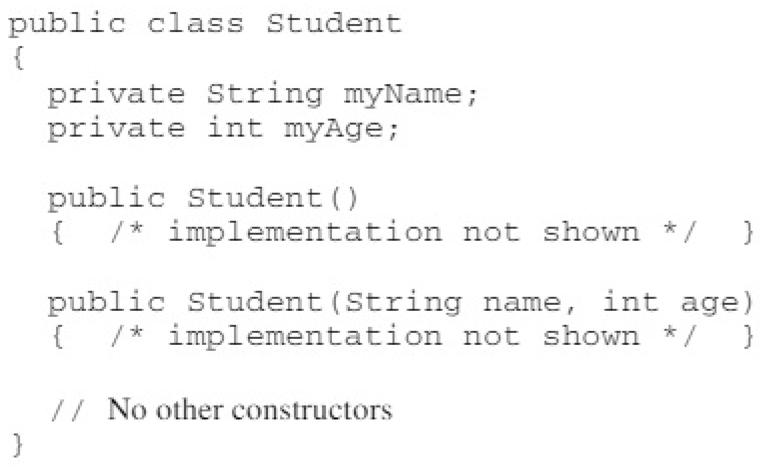
Ⓑ student.myName = "Tom";

Ⓒ student.getName ("Tom");

Ⓓ student.setName ("Tom");

Ⓔ Person.setName ("Tom");

18. Consider the following class declaration.



```
public class Student
{
  private String myName;
  private int myAge;

  public Student()
  {  /* implementation not shown */  }

  public Student(String name, int age)
  {  /* implementation not shown */  }

  // No other constructors
}
```

Which of the following declarations will compile without error?
1.
Student a = new Student();
2.
Student b = new Student("Juan", 15);
3.
Student c = new Student("Juan", "15");

U1-4_2

Ⓐ I only

Ⓑ II only

Ⓒ I and II only

Ⓓ I and III only

Ⓔ I, II, and III

19. Consider the following class that stores information about temperature readings on various dates.

```
public class TemperatureReading implements Comparable
{
private double temperature; private int month, day, year;
public int compareTo(Object obj)
{
TemperatureReading other = (TemperatureReading) obj;
/* missing code */
}
// There may be instance variables, constructors, and methods that are not shown.
}
```

Consider the following code segments that are potential replacements for /* *missing code* */.

1.

```
Double d1 = new Double(temperature); Double d2 = new Double(other.temperature);
return d1.compareTo(d2);
```

2.

```
if (temperature < other.temperature)
   return -1;
else if (temperature == other.temperature)
   return 0;
else
   return 1;
```

III. return (int) (temperature - other.temperature);

Which of the code segments could be used to replace /* *missing code* */ so that compareTo can be used to order TemperatureReading objects by increasing temperature value?

Ⓐ II only

Ⓑ I and II only

Ⓒ I and III only

Ⓓ II and III only

Ⓔ I, II, and III

---

20. Consider the following class.

```
public class SomeMethods
{
public void one(int first)
{ / * implementation not shown * / }
public void one(int first, int second)
{ / * implementation not shown * / }
public void one(int first, String second)
{ / * implementation not shown * / }
}
```

Which of the following methods can be added to the SomeMethods class without causing a compile-time error?

1.

```
public void one(int value)
{ / * implementation not shown * / }
```

2.

```
public void one (String first, int second
{ / * implementation not shown * / }
```

3.

```
public void one (int first, int second, int third)
{ / * implementation not shown * / }
```

U1-4_2

Ⓐ I only

Ⓑ I and II only

Ⓒ I and III only

Ⓓ II and III only

Ⓔ I, II, and III