

Started on Monday, 16 May 2022, 12:44 PM

State Finished

Completed on Monday, 16 May 2022, 1:05 PM

Time taken 21 mins 4 secs

Question 1

Complete

Marked out of
1.00

Given the following my.school.ClassRoom and my.city.School class definitions, which line numbers in main() generate a compiler error? (Choose all that apply)

```
1: package my.school;
2: public class Classroom {
3:     private int roomNumber;
4:     protected String teacherName;
5:     static int globalKey = 54321;
6:     public int floor = 3;
7:     Classroom(int r, String t) {
8:         roomNumber = r;
9:         teacherName = t; } }
1: package my.city;
2: import my.school.*;
3: public class School {
4:     public static void main(String[] args) {
5:         System.out.println(Classroom.globalKey);
6:         Classroom room = new Classroom(101, ""Mrs. Anderson");
7:         System.out.println(room.roomNumber);
8:         System.out.println(room.floor);
9:         System.out.println(room.teacherName); } }
```

Select one or more:

- ☒ **a. None, the code compiles fine.**
- ☐ **b. Line 8**
- ☐ **c. Line 5**
- ☐ **d. Line 9**
- ☐ **e. Line 6**
- ☐ **f. Line 7**

Question

2

Complete

Marked out of
1.00

Which are the results of the following code? (Choose all that apply)

```
String letters = "abcdef";
```

```
System.out.println(letters.length());
```

```
System.out.println(letters.charAt(3));
```

```
System.out.println(letters.charAt(6));
```

Select one or more:

- ☐ a. c
- ☒ b. An exception is thrown.
- ☐ c. d
- ☐ d. 5
- ☒ e. The code does not compile.
- ☐ f. 6

Question

3

Complete

Marked out of
1.00

Which are true statements? (Choose all that apply)

Select one or more:

- ☐ a. **StringBuilder** is immutable.
- ☒ b. **StringBuffer** is immutable.
- ☐ c. An immutable object cannot be garbage collected.
- ☒ d. **String** is immutable.
- ☒ e. An immutable object can be garbage collected.
- ☐ f. An immutable object can be modified.
- ☒ g. An immutable object cannot be modified.

Question

4

Complete

Marked out of
1.00

What is the output of the following code?

```
1: package rope;  
2: public class Rope {  
3:     public static int LENGTH = 5;  
4:     static {  
5:         LENGTH = 10;  
6:     }  
7:     public static void swing() {  
8:         System.out.print("swing ");  
9:     }  
10: }  
  
1: import rope.*;  
2: import static rope.Rope.*;  
3: public class Chimp {  
4:     public static void main(String[] args) {  
5:         Rope.swing();  
6:         new Rope().swing();  
7:         System.out.println(LENGTH);  
8:     }  
9: }
```

Select one or more:

- ☐ a. Compiler error on line 2 of Chimp.
- ☐ b. Compiler error on line 7 of Chimp.
- ☐ c. **swing swing 10**
- ☒ d. Compiler error on line 6 of Chimp.
- ☐ e. swing swing 5
- ☐ f. Compiler error on line 5 of Chimp.

Question

5

Complete

Marked out of
1.00

Which of the following are true? (Choose all that apply)

Select one or more:

- ☐ a. A public class that has private fields and package private methods is not visible to classes outside the package.
- ☐ b. You can use access modifiers to restrict read access to all classes that begin with the word Test.
- ☐ c. You can use access modifiers to allow read access to all methods, but not any instance variables.
- ☒ d. **You can use access modifiers so only some of the classes in a package see a particular package private class.**
- ☐ e. Package private access is more lenient than protected access.

Question 6

Complete

Marked out of 1.00

Which are methods using JavaBeans naming conventions for accessors and mutators?
(Choose all that apply)

Select one or more:

- ☒ a. `public int getNumWings() { return numberWings;}`
- ☐ b. `public boolean canSwim() { return numberWings;}`
- ☐ c. `public int numWings() { return numberWings;}`
- ☒ d. `public boolean getCanSwim() { return canSwim;}`
- ☒ e. `public void setCanSwim(boolean b) { canSwim = b;}`

Question 7

Complete

Marked out of 1.00

Which of the following are true? (Choose all that apply)

Select one or more:

- ☐ a. Encapsulation uses private instance variables.
- ☒ b. Encapsulation allows setters.
- ☐ c. Immutability allows setters.
- ☐ d. Immutability uses private instance variables.
- ☐ e. Immutability uses package private instance variables.
- ☐ f. Encapsulation uses package private instance variables.

Question 8

Complete

Marked out of 1.00

What is the result of the following code? (Choose all that apply)

13: String a = "";

14: a += 2;

15: a += 'c';

16: a += false;

17: if (a == "2cfalse") System.out.println("==");

18: if (a.equals("2cfalse")) System.out.println("equals");

Select one or more:

- ☐ a. An exception is thrown.
- ☐ b. Compile error on another line.
- ☐ c. ==
- ☐ d. Compile error on line 15.
- ☒ e. equals
- ☐ f. Compile error on line 16.
- ☐ g. Compile error on line 14.

Question

9

Complete

Marked out of
1.00

Which statement(s) are correct about the following code? (Choose all that apply)

```
public class Rodent {  
    protected static Integer chew() throws Exception {  
        System.out.println("Rodent is chewing");  
        return 1;  
    }  
}  
  
public class Beaver extends Rodent {  
    public Number chew() throws RuntimeException {  
        System.out.println("Beaver is chewing on wood");  
        return 2;  
    }  
}
```

Select one or more:

- ☐ a. It fails to compile because the method is protected in the parent class and public in the subclass.
- ☐ b. It fails to compile because the type of the exception the method throws is a subclass of the type of exception the parent method throws.
- ☐ c. It fails to compile because the return types are not covariant.
- ☐ d. It will compile without issue.
- ☒ e. It fails to compile because of a static modifier mismatch between the two methods.

Question

10

Complete

Marked out of
1.00

Which are the results of the following code? (Choose all that apply)

```
String numbers = "012345678";  
System.out.println(numbers.substring(1, 3));  
System.out.println(numbers.substring(7, 7));  
System.out.println(numbers.substring(7));
```

Select one or more:

- ☐ a. 7
- ☐ b. 123
- ☒ c. 78
- ☒ d. 12
- ☐ e. A blank line.
- ☐ f. An exception is thrown.
- ☐ g. The code does not compile.

Question 11

Complete

Marked out of 1.00

Which of the following statements can be inserted in the blank line so that the code will compile successfully? (Choose all that apply)

```
public interface CanHop {}

public class Frog implements CanHop {

    public static void main(String[] args) {

        frog = new TurtleFrog();

    }

}

public class BrazilianHornedFrog extends Frog {}

public class TurtleFrog extends Frog {}
```

Select one or more:

- ☐ a. **BrazilianHornedFrog**
- ☐ b. **Long**
- ☐ c. **Frog**
- ☒ d. **TurtleFrog**
- ☐ e. **CanHop**
- ☐ f. **Object**

Question 12

Complete

Marked out of 1.00

Which of the following may only be hidden and not overridden? (Choose all that apply)

Select one or more:

- ☒ a. **protected instance methods**
- ☐ b. **public variables**
- ☐ c. **public instance methods**
- ☐ d. **static methods**
- ☐ e. **private variables**
- ☒ f. **private instance methods**

Question 13

Complete

Marked out of 1.00

What modifiers are implicitly applied to all interface methods? (Choose all that apply)

Select one or more:

- ☐ a. **static**
- ☒ b. **abstract**
- ☐ c. **default**
- ☒ d. **protected**
- ☐ e. **public**
- ☐ f. **void**

Question 14

Complete

Marked out of 1.00

What is the output of the following code?

```
1: class Mammal {  
2: public Mammal(int age) {  
3: System.out.print("Mammal");  
4: }  
5: }  
6: public class Platypus extends Mammal {  
7: public Platypus() {  
8: System.out.print("Platypus");  
9: }  
10: public static void main(String[] args) {  
11: new Mammal(5);  
12: }  
13: }
```

Select one or more:

- ☐ a. The code will not compile because of line 8.
- ☐ b. MammalPlatypus
- ☐ c. Mammal
- ☒ d. The code will not compile because of line 11.
- ☐ e. Platypus
- ☐ f. PlatypusMammal

Question 15

Complete

Marked out of 1.00

What is the result of the following code?

```
3: String s = "purr";  
4: s.toUpperCase();  
5: s.trim();  
6: s.substring(1, 3);  
7: s += " two";  
8: System.out.println(s.length());
```

Select one or more:

- ☐ a. 10
- ☒ b. 8
- ☐ c. The code does not compile.
- ☐ d. 2
- ☐ e. An exception is thrown.
- ☐ f. 4

