Started on	Saturday, 9 July 2022, 5:06 PM
State	Finished
Completed on	Saturday, 9 July 2022, 5:13 PM
Time taken	7 mins 5 secs
Marks	4.00/10.00
Grade	40.00 out of 100.00

Question 1

Partially correct

Mark 0.50 out of 1.00

Which statements are true about nested classes in Java?

Select one or more:

- a. a nested class is not a member of its enclosing class
- b. the scope of a nested class is bounded by the scope of its enclosing class
- c. a nested class has access to the members, including private members, of the class in which it is nested ✔
- d. a nested class has not access to the private members of the class in which it is nested
- e. as a member of its enclosing class, a nested class cannot be declared as private ✖

Your answer is partially correct.

You have correctly selected 1.

The correct answers are: the scope of a nested class is bounded by the scope of its enclosing class, a nested class has access to the members, including private members, of the class in which it is nested

Question 2

Incorrect

Mark 0.00 out of 1.00

Suppose we have next **Outer** and non-static **Inner** classes:

```
class Outer {
    public Outer(String message) {
        System.out.println(message);
    }
    class Inner {
        public Inner(String message) {
            System.out.println(message);
        }
    }
}
```

Which of the following correctly creates an instance of both the **Outer** and the **Inner** classes?

Select one:

a.
```
Outer outer = new Outer("Inside Outer class");
Inner inner = outer.new Inner("Inside Inner class");
```
✖

b.
```
Outer outer = new Outer("Inside Outer class");
Outer.Inner inner = Outer.new Inner("Inside Inner class");
```

c.
```
Outer outer = new Outer("Inside Outer class");
Outer.Inner inner = new Inner("Inside Inner class");
```

d.
```
Outer outer = new Outer("Inside Outer class");
Outer.Inner inner = outer.new Inner("Inside Inner class");
```

Your answer is incorrect.

The correct answer is:
```
Outer outer = new Outer("Inside Outer class");
Outer.Inner inner = outer.new Inner("Inside Inner class");
```

Question 3

Partially correct

Mark 0.50 out of 1.00

Which statements are true about a static nested class?

Select one or more:

- a. it can be accessed without instantiating the outer class
- b. it must extend the enclosing class
- c. you must have a reference to an instance of the enclosing class in order to instantiate it
- d. it does not have access to nonstatic members of the enclosing class ✔
- e. it's variables and methods must be static

Your answer is partially correct.

You have correctly selected 1.

The correct answers are: it does not have access to nonstatic members of the enclosing class, it can be accessed without instantiating the outer class

Question **4**

Incorrect

Mark 0.00 out of 1.00

Suppose we have next **Outer** and static **Inner** classes:

```
class Outer {
    public Outer(String message) {
        System.out.println(message);
    }
    static class Inner {
        public Inner(String message) {
            System.out.println(message);
        }
    }
}
```

Which of the following correctly creates an instance of both the **Outer** and the **Inner** classes?

Select one:

a.
```
Outer outer = new Outer("Inside Outer class");
Outer.Inner inner = outer.new Inner("Inside Inner class");
```

b.
```
Outer outer = new Outer("Inside Outer class");
Outer.Inner inner = Outer.new Inner("Inside Inner class");
```

c.
```
Outer outer = new Outer("Inside Outer class");
Outer.Inner inner = new Outer.Inner("Inside Inner class");
```

d. ✖
```
Outer outer = new Outer("Inside Outer class");
Outer.Inner inner = new Inner("Inside Inner class");
```

Your answer is incorrect.

The correct answer is:
```
Outer outer = new Outer("Inside Outer class");
Outer.Inner inner = new Outer.Inner("Inside Inner class");
```

Question 5

Partially correct

Mark 0.33 out of 1.00

Which statements are true about a local class?

Select one or more:

- [ ] a. it cannot access the local variables of the enclosing method unless they are final or effectively final
- [ ] b. it can be instantiated only within the method where it is defined
- [ ] c. it can be instantiated outside the method where it is defined
- [x] d. it cannot be declared with private access modifier ✔
- [ ] e. it must extend the enclosing class

Your answer is partially correct.

You have correctly selected 1.
The correct answers are: it can be instantiated only within the method where it is defined, it cannot access the local variables of the enclosing method unless they are final or effectively final, it cannot be declared with private access modifier

Question 6

Incorrect

Mark 0.00 out of 1.00

Suppose we have next **Main** and **Local** classes:

```
1  public class Main {
2    public static void main(String[] args) {
3      int multiplier = 10;
4      class Local {
5        int multiply(int number) {
6          return number * multiplier;
7        }
8      }
9      multiplier = 5;
10     int result = new Local().multiply(20);
11     System.out.println(result);
12   }
13 }
```

What is the result of executing the program given above:

Select one:

- a. 200
- b. 100 ✖
- c. 0
- d. compile error in line 6

Your answer is incorrect.

The correct answer is: compile error in line 6

Question 7

Correct

Mark 1.00 out of 1.00

Suppose we have the next **Main** class:

```
2   class Main {
3     public static void main(String[] args) {
4       Object object1 = new Object() {
5         public boolean equals(Object obj) {
6           return this == obj;
7         }
8       };
9       Object object2 = object1;
10      System.out.println(object1.equals(object2));
11    }
12  }
```

What is the result of executing the program given above:

Select one:

- a. compile error in line 4
- b. true ✔
- c. false
- d. runtime error

Your answer is correct.

The correct answer is: true

Question 8

Correct

Mark 1.00 out of 1.00

Suppose we have the next abstract class:

```
1   abstract class Shape {
2     protected String color;
3     public Shape(String color) {
4       this.color = color;
5     }
6     abstract public void draw();
7   }
```

What is the result of executing the following program:

```
9   public class Main {
10    public static void main(String[] args) {
11      Shape shape = new Shape() {
12        String color = "Green";
13          public void draw() {
14            System.out.println("Shape is " + color);
15          }
16        };
17        shape.draw();
18    }
19  }
```

Select one:

- a. Shape is null
- b. compile error in line 12
- c. compile error in line 11 ✔
- d. Shape is Green

Your answer is correct.

The correct answer is: compile error in line 11

Question 9

Partially correct

Mark 0.67 out of 1.00

Which statements are true about the **enum** type?

Select one or more:

- [x] a. an enum type extend java.lang.Enum class ✔
- [ ] b. an enum type implement java.lang.Comparable interface
- [ ] c. an enum cannot have a constructor
- [ ] d. it can be declared only inside another class
- [x] e. it can be declared inside or outside any non-static class ✔

Your answer is partially correct.

You have correctly selected 2.
The correct answers are: it can be declared inside or outside any non-static class, an enum type extend java.lang.Enum class, an enum type implement java.lang.Comparable interface

Question 10

Incorrect

Mark 0.00 out of 1.00

Suppose we have the next enum:

```
1  enum Directions {
2      NORTH(1), SOUTH(2), WEST(3), EAST(4);
3      public int value;
4      public Directions(int number) {
5          this.value = number;
6      }
7  }
```

What is the result of executing the following program:

```
9  public class Main {
10   public static void main(String[] args) {
11     for (Directions direction: Directions.values()) {
12       System.out.print(direction.value + " ");
13     }
14   }
15 }
```

Select one:

- a. compile error in line 4
- b. NORTH SOUTH WEST EAST ✖
- c. 1 2 3 4
- d. compile error in line 2

Your answer is incorrect.

The correct answer is: compile error in line 4

◄ REPO. INNER, OUTER CLASS. ENUM

Jump to...

USEFUL LINKS ►