OCA / OCP Java SE 8 Programmer Practice Tests

PREV
Chapter 5 Using Loop Constructs

NEXT
Chapter 7 Working with Inheritance

Chapter 6
Working with Methods and Encapsulation

THE OCA EXAM TOPICS COVERED IN THIS PRACTICE TEST INCLUDE THE FOLLOWING:

- ✓**Working with Methods and Encapsulation**
- Create methods with arguments and return values; including overloaded methods
- Apply the static keyword to methods and fields
- Create and overload constructors; differentiate between default and user defined constructors
- Apply access modifiers
- Apply encapsulation principles to a class
- Determine the effect upon object references and primitive values when they are passed into methods that change the values

1. Fill in the blanks: The______________ access modifier allows access to everything the ______________access modifier does and more.

1. package-private, `protected`
2. `protected`, `public`
3. `protected`, package-private
4. `private`, package-private

2. What is the command to call one constructor from another constructor in the same class?

1. `super()`
2. `this()`
3. `that()`
4. `construct()`

3. What is the output of the following application?

```
package stocks;
public class Bond {
   private static int price = 5;
   public boolean sell() {
      if(price<10) {
         price++;
         return true;
      } else if(price>=10) {
         return false;
      new Bond().sell();
      new Bond().sell();
      System.out.print(price);
   }
}
```

1. 5
2. 6
3. 8
4. The code does not compile.

4. What is true about the following program?

```
package figures;
public class Dolls {
   public void nested() { nested(2,true); } // g1
   public int nested(int level, boolean height) { return nested(lev
   public int nested(int level) { return level+1; }; // g2

   public static void main(String[] outOfTheBox) {
      System.out.print(new Dolls().nested());
   }
}
```

1. It compiles successfully and prints 3 at runtime.
2. It does not compile because of line g1.
3. It does not compile because of line g2.
4. It does not compile for some other reason.

5. Fill in the blank: Java uses ____________ to send data into a method.

1. pass-by-null
2. pass-by-value
3. both pass-by-value and pass-by-reference
4. pass-by-reference

6. Which of the following is a valid JavaBean method signature?

1. `public void getArrow()`
2. `public void setBow()`
3. `public void setRange(int range)`
4. `public String addTarget(String target)`

7. Which of the following statements about calling `this()` in a constructor is not true?

1. If `this()` is used, it must be the first line of the constructor.
2. If `super()` and `this()` are both used in the same constructor, `super()` must appear on the line immediately after `this()`.
3. If arguments are provided to `this()`, then there must be a constructor in the class able to take those arguments.
4. If the no-argument `this()` is called, then the class must explicitly implement the no-argument constructor.

8. Which of the following can fill in the blank to make the class compile?

```
package ai;
public class Robot {
   ___________ compute() { return 10; }
}
```

1. `Public int`
2. `Long`
3. `void`
4. `private String`

9. Fill in the blank: A ______________variable is always available to all instances of the class.

1. `public`
2. `local`
3. `static`
4. instance

10. Which line of code, inserted at line `p1`, causes the application to print `5`?

```
package games;
public class Jump {
   private int rope = 1;
   protected boolean outside;
   public Jump() {
      // p1
      outside = true;
   }
   public Jump(int rope) {
      this.rope = outside ? rope : rope+1;
   }
   public static void main(String[] bounce) {
      System.out.print(new Jump().rope);
   }
}
```

1. `this(4);`
2. `new Jump(4);`
3. `this(5);`
4. `rope = 4;`

11. Which of the following statements is not true?

1. An instance of one class may access an instance of another class's attributes if it has a reference to the instance and the attributes are declared `public`.
2. An instance of one class may access package-private attributes in a parent class, provided the parent class is not in the same package.
3. Two instances of the same class may access each other's `private` attributes.
4. An instance of one class may access an instance of another class's attributes if both classes are located in the same package and marked `protected`.

12. Given the following class, what should be inserted into the two blanks to ensure the class data is properly encapsulated?

```
package storage;
public class Box {
   public String stuff;
    ___________String___________ () {
      return stuff;
   }

   public void setStuff(String stuff) {
      this.stuff = stuff;
   }
}
```

1. `public` and `getStuff`
2. `private` and `isStuff`
3. `public` and `setStuff`
4. None of the above

13. Which statement about a no-argument constructor is true?

1. The Java compiler will always insert a default no-argument constructor if you do not define a no-argument constructor in your class.
2. In order for a class to call `super()` in one of its constructors, its parent class must explicitly implement a no-argument constructor.
3. If a class extends another class that has only one constructor that takes a value, then the child class must explicitly declare at least one constructor.

4. A class may contain more than one no-argument constructor.

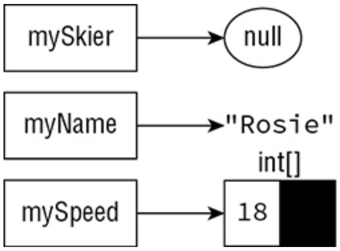
14. Which of the following method signatures does not contain a compiler error?

1. `public void sing(String key, String... harmonies)`
2. `public void sing(int note, String... sound, int music)`
3. `public void sing(String... keys, String... pitches)`
4. `public void sing(String... notes, String melodies)`

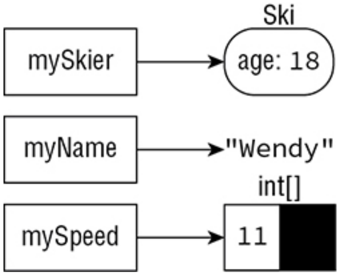
15. Given the following application, which diagram best represents the state of the `mySkier`, `mySpeed`, and `myName` variables in the `main()` method after the call to the `slalom()` method?

```
package slopes;
public class Ski {
   private int age = 18;
   private static void slalom(Ski racer, int[] speed, String name)
      racer.age = 18;
      name = "Wendy";
      speed = new int[1];
      speed[0] = 11;
      racer = null;
   }
   public static void main(String... mountain) {
      final Ski mySkier = new Ski();
      mySkier.age = 16;
      final int[] mySpeed = new int[1];
      final String myName = "Rosie";
      slalom(mySkier,mySpeed,myName);
   }
}
```

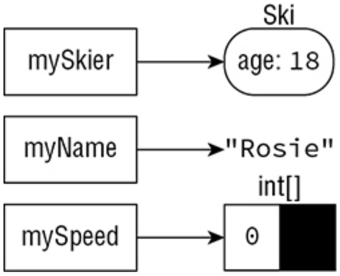
1.



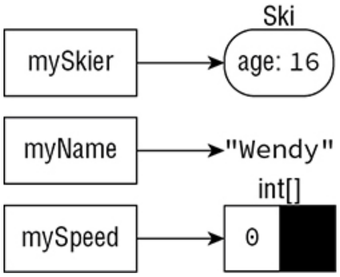
2.



3.



4.



16. Given the class below, which method signature could be successfully added to the class as an overloaded version of the `findAverage()` method?

```
public class Calculations {
   public Integer findAverage(int sum) { return sum; }
}
```

1. `public Long findAverage(int sum)`
2. `public Long findAverage(int sum, int divisor)`
3. `public Integer average(int sum)`
4. `private void findAverage(int sum)`

17. Which of the following is not a reason to use encapsulation when designing a class?

1. Promote usability by other developers.
2. Maintain class data integrity of data elements.
3. Prevent users from modifying the internal attributes of a class.
4. Increase concurrency and improve performance.

18. Which of the following data types can be modified after they are passed to a method as an argument?

1. `int[]`
2. `String`
3. `long`
4. `boolean`

19. What is the best way to call the following method from another class in the same package, assuming the class using the method does not have any `static` imports?

```
package useful;
public class MathHelper {
   public static int roundValue(double d) {
      // Implementation omitted
   }
}
```

1. `MathHelper:roundValue(5.92)`
2. `MathHelper.roundValue(3.1)`
3. `roundValue(4.1)`
4. `useful.MathHelper.roundValue(65.3)`

20. Given a method with one of the following return types, which data type prevents the `return` statement from being used within the method?

1. `byte`
2. `String`
3. `void`
4. None of the above

21. How many `final` modifiers would need to be removed for this application to compile?

```
package end;
public final class Games {
   public final static int finish(final int score) {
      final int win = 3;
      final int result = score++ < 5 ? 2 : win;
      return result+=win;
   }
   public static void main(final String[] v) {
      System.out.print(finish(Integer.parseInt(v[0])));
   }
}
```

1. None
2. One
3. Two
4. The code will not compile regardless of the number of `final` modifiers that are removed.

22. Fill in the blanks: ______________is used to call a constructor in the parent class, while ______________is used to reference a member of the parent class.

1. `super` and `this()`
2. `super` and `super()`
3. `super()` and `this`
4. `super()` and `super`

23. Given the following method signature, which classes can call it?

```
void run(String government)
```

1. Classes in other packages
2. Classes in the same package
3. Subclasses in a different package
4. All classes

24. Which statement(s) about the following class would help to properly encapsulate the data in the class?

```
package shield;
public class Protect {
   private String material;
   protected int strength;

   public int getStrength() {
      return strength;
   }
   public void setStrength(int strength) {
      this.strength = strength;
   }
}
```

1. Change the access modifier of `strength` to `private`.
2. Add a getter method for `material`.
3. Add a setter method for `material`.

1. I
2. II and III
3. I, II, and III
4. None, the data in the class is already encapsulated.

25. Which of the following is a valid method name in Java?

1. `Go_$Outside$2()`
2. `have-Fun()`
3. `new()`
4. `9enjoyTheWeather()`

26. Which of the following lines of code can be inserted in the line below that would allow the class to compile?

```
package farm;
public class Coop {
   public final int static getNumberOfChickens() {
      // INSERT CODE HERE
   }
}
```

1. `return 3.0;`
2. `return 5L;`
3. `return 10;`
4. None of the above

27. Which of the following is a true statement about passing data to a method?

1. A change made to a primitive value passed to a method is reflected in the calling method.
2. A change made to the data within an object passed to a method is reflected in the calling method.
3. Reassigning an object reference passed to a method is reflected in the calling method.
4. A change made to a `boolean` value passed to a method is reflected in the calling method.

28. What is a possible output of the following application?

```
package wrap;
public class Gift {
   private final Object contents;
   protected Object getContents() {
      return contents;
   }
   protected void setContents(Object contents) {
      this.contents = contents;
   }
   public void showPresent() {
      System.out.print("Your gift: "+contents);
   }
   public static void main(String[] treats) {
      Gift gift = new Gift();
      gift.setContents(gift);
      gift.showPresent();
   }
}
```

1. `Your gift: wrap.Gift@29ca2745`
2. `Your gift: Your gift:`
3. It does not compile.
4. It compiles but throws an exception at runtime.

29. Which of the following is a valid JavaBean method prefix?

1. `is`
2. `gimme`
3. `request`
4. `put`

30. Given the following two classes, each in a different package, which line inserted below allows the second class to compile?

```
package clothes;
public class Store {
   public static String getClothes() { return "dress"; }
}

package wardrobe;
// INSERT CODE HERE
public class Closet {
   public void borrow() {
      System.out.print("Borrowing clothes: "+getClothes());
   }
}
```

1. `static import clothes.Store.getClothes;`
2. `import clothes.Store.*;`
3. `import static clothes.Store.getClothes;`
4. `import static clothes.Store;`

31. What access modifier is used to mark class members package-private?

    1. `private`
    2. `default`
    3. `protected`
    4. None of the above

32. How many lines of the following program contain compilation errors?

```
package sky;
public class Stars {
   private int inThe = 4;
   public void Stars() {
      super();
   }
   public Stars(int inThe) {
      this.inThe = this.inThe;
   }
   public static void main(String[] endless) {
      System.out.print(new sky.Stars(2).inThe);
   }
}
```

    1. `None`
    2. One
    3. Two
    4. Three

33. Which of the following statements is true?

    1. An instance method is allowed to reference a `static` variable.
    2. A `static` method is allowed to reference an instance variable.
    3. A `static` initialization block is allowed to reference an instance variable.
    4. A `final static` variable may be set in a constructor.

34. Given the following method declaration, which line can be inserted to make the code compile?

```
public short calculateDistance(double lat1, double lon1,
      double lat2, double lon2) {
   // INSERT CODE HERE
}
```

    1. `return new Integer(3);`
    2. `return new Byte((byte)6);`
    3. `return 5L;`
    4. `return new Short(4).longValue();`

35. Which of the following statements about overloaded methods are true?

    1. Overloaded methods must have the same name.
    2. Overloaded methods must have the same return type.
    3. Overloaded methods must have a different list of parameters.

    1. I
    2. I and II
    3. I and III
    4. I, II, and III

36. How many lines of code would need to be removed for the following class to compile?

```
package work;
public class Week {
   private static final String monday;
   String tuesday;
   final static wednesday = 3;
   final protected int thursday = 4;
}
```

    1. One

2. Two
3. Three
4. The code will not compile regardless of the number of lines removed.

37. What is the output of the following application?

```
package pet;
public class Puppy {
   public static int wag = 5;   // q1
   public void Puppy(int wag) { // q2
      this.wag = wag;
   }
   public static void main(String[] tail) {
      System.out.print(new Puppy(2).wag); // q3
   }
}
```

1. `2`
2. It does not compile because of line `q1`.
3. It does not compile because of line `q2`.
4. It does not compile because of line `q3`.

38. Fill in the blanks: The ______________access modifier allows access to everything the ______________access modifier does and more.

1. `public`, `private`
2. `private`, package-private
3. package-private, `protected`
4. `private`, `public`

39. What is the output of the following application?

```
package ship;
public class Phone {
   private int size;
   public Phone(int size) {this.size=size;}

   public static void sendHome(Phone p, int newSize) {
      p = new Phone(newSize);
      p.size = 4;
   }
   public static final void main(String... params) {
      final Phone phone = new Phone(3);
      sendHome(phone,7);
      System.out.print(phone.size);
   }
}
```

1. `3`
2. `4`
3. `7`
4. The code does not compile.

40. Given the following class, which line of code when inserted below would prevent the class from compiling?

```
public class Drink {
   public static void water() {}
   public void get() {
      // INSERT CODE HERE
   }
}
```

1. `water();`
2. `this.Drink.water();`
3. `this.water();`
4. `Drink.water();`

41. Given the following method declaration signature, which of the following is a valid call of this method?

```
public void call(int count, String me, String... data)
```

1. `call(9,"me",10,"Al")`

2. `call(5)`
3. `call(2,"home","sweet")`
4. `call("answering","service")`

42. Which statement about a `static` variable is true?

1. The value of a `static` variable must be set when the variable is declared or in a `static` initialization block.
2. It is not possible to read `static final` variables outside the class in which they are defined.
3. It is not possible to reference `static` methods using `static` imports.
4. A `static` variable is always available in all instances of the class.

43. Which of the following is not a true statement?

1. The first line of every constructor is a call to the parent constructor via the `super()` command.
2. A class does not have to have a constructor explicitly defined.
3. A constructor may pass arguments to the parent constructor.
4. A `final` instance variable whose value is not set when they are declared or in an initialization block should be set by the constructor.

44. How many `final` modifiers would need to be removed for this application to compile?

```
package park;
public class Tree {
   public final static long numberOfTrees;
   public final double height;
   static {}
   { final int initHeight = 2;
     height = initHeight;
   }
   static {
      numberOfTrees = 100;
      height = 4;
   }
}
```

1. None
2. One
3. Two
4. The code will not compile regardless of the number of `final` modifiers removed.

45. What is the output of the following application?

```
package jungle;
public class RainForest extends Forest {
   public RainForest(long treeCount) {
      this.treeCount = treeCount+1;
   }
   public static void main(String[] birds) {
      System.out.print(new RainForest(5).treeCount);
   }
}
class Forest {
   public long treeCount;
   public Forest(long treeCount) {
      this.treeCount = treeCount+2;
   }
}
```

1. `5`
2. `6`
3. `8`
4. The code does not compile.

46. What is the output of the following application?

```
public class ChooseWisely {
   public ChooseWisely() { super(); }
   public int choose(int choice) { return 5; }
   public int choose(short choice) { return 2; }
   public int choose(long choice) { return 11; }
   public static void main(String[] path) {
      System.out.print(new ChooseWisely().choose((byte)2+1));
      }
   }
```

1. 5
2. 2
3. 11
4. The code does not compile.

47. What is the output of the following application?

```
   package sports;
   public class Football {
      public static Long getScore(Long timeRemaining) {
         return 2*timeRemaining; // m1
      }
      public static void main(String[] refs) {
         final int startTime = 4;
         System.out.print(getScore(startTime)); // m2
      }
   }
```

1. 8
2. The code does not compile because of line `m1`.
3. The code does not compile because of line `m2`.
4. The code compiles but throws an exception at runtime.

48. Which of the following is a valid method name in Java?

1. `$sprint()`
2. `\jog13()`
3. `walk#()`
4. `%run()`

49. Assume there is a class `Bouncer` with a `protected` variable. Methods in which class can access this variable?

1. Only subclasses of `Bouncer`
2. Any subclass of `Bouncer` or any class in the same package as `Bouncer`
3. Only classes in the same package as `Bouncer`
4. Any superclass of `Bouncer`

50. Given the following two classes, each in a different package, which line inserted below allows the second class to compile?

```
   package commerce;
   public class Bank {
      public void withdrawal(int amountInCents) {}
      public void deposit(int amountInCents) {}
   }

   package employee;
   // INSERT CODE HERE
   public class Teller {
      public void processAccount(int depositSlip, int withdrawalSlip)
         withdrawal(withdrawalSlip);
         deposit(depositSlip);
      }
   }
```

1. `import static commerce.Bank.*;`
2. `static import commerce.Bank.*;`
3. `import static commerce.Bank;`
4. None of the above

PREV

NEXT

Chapter 5 Using Loop Constructs

Chapter 7 Working with Inheritance