

unit5_final_review

1. A rectangular box fits inside another rectangular box if and only if the height, width, and depth of the smaller box are each less than the corresponding values of the larger box. Consider the following three interface declarations that are intended to represent information necessary for rectangular boxes.

I.

```
public interface RBox
{
    /** @return the height of this RBox */
    double getHeight();

    /** @return the width of this RBox */
    double getWidth();

    /** @return the depth of this RBox */
    double getDepth();
}
```

II.

```
public interface RBox
{
    /** @return true if the height of this RBox is less than the height of other;
     *      false otherwise
     */
    boolean smallerHeight(RBox other);

    /** @return true if the width of this RBox is less than the width of other;
     *      false otherwise
     */
    boolean smallerWidth(RBox other);

    /** @return true if the depth of this RBox is less than the depth of other;
     *      false otherwise
     */
    boolean smallerDepth(RBox other);
}
```

III.

```
public interface RBox
{
    /** @return the surface area of this RBox */
    double getSurfaceArea();

    /** @return the volume of this RBox */
    double getVolume();
}
```

Which of the interfaces, if correctly implemented by a Box class, would be sufficient functionality for a user of the Box class to determine if one Box can fit inside another?

- (A) I only
- (B) II only
- (C) III only
- (D) I and II only
- (E) I, II, and III

unit5_final_review

2. The following question is based on the following incomplete declaration of the class `BoundedIntArray` and its constructor definitions.

A `BoundedIntArray` represents an indexed list of integers. In a `BoundedIntArray` the user can specify a size, in which case the indices range from 0 to `size - 1`. The user can also specify the lowest index, `low`, in which case the indices can range from `low` to `low + size - 1`.

```
public class BoundedIntArray
{
    private int[] myItems; // storage for the list

    private int myLowIndex; // lowest index

    public BoundedIntArray(int size)
    {
        myItems = new int[size];
        myLowIndex = 0;
    }

    public BoundedIntArray(int size, int low)
    {
        myItems = new int[size];
        myLowIndex = low;
    }

    // other methods not shown
}
```

Consider the following statements.

```
BoundedIntArray arr1 = new BoundedIntArray(100, 5);
```

unit5_final_review

```
BoundedIntArray arr2 = new BoundedIntArray(100);
```

Which of the following best describes `arr1` and `arr2` after these statements?

- (A) `arr1` and `arr2` both represent lists of integers indexed from 0 to 99.
- (B) `arr1` and `arr2` both represent lists of integers indexed from 5 to 104.
- (C) `arr1` represents a list of integers indexed from 0 to 104, and `arr2` represents a list of integers indexed from 0 to 99.
- (D) `arr1` represents a list of integers indexed from 5 to 99, and `arr2` represents a list of integers indexed from 0 to 99.
- (E) `arr1` represents a list of integers indexed from 5 to 104, and `arr2` represents a list of integers indexed from 0 to 99.

3. Consider the following class definition.

```
public class Box
{
    private double weight;

    /** Postcondition: weight is initialized to w. */
    public Box(double w)
    {
        /* implementation not shown */
    }

    public double getWeight()
    {
        return weight;
    }

    public void addWeight(double aw)
    {
        /* missing statement */
    }
}
```

The following code segment, which appears in a class other than `Box`, is intended to create a `Box` object `b1` with a weight of 2.2 units and then increase the weight of `b1` by 1.5 units.

```
Box b1 = new Box(2.2);
b1.addWeight(1.5);
```

Which of the following statements could replace `/* missing statement */` so that the code segment works as intended?

unit5_final_review

- (A) `aw += weight;`
- (B) `aw += getWeight();`
- (C) `weight += aw;`
- (D) `weight += getWeight();`
- (E) `return weight + aw;`

4. Consider the following class declaration.

```
public class Sample
{
    private int a;
    private double b;

    public Sample(int x, double y)
    {
        a = x;
        b = y;
    }

    // No other constructors
}
```

The following method appears in a class other than `Sample`.

```
public static void test()
{
    Sample object = new /* missing constructor call */ ;
}
```

Which of the following could be used to replace `/* missing constructor call */` so that the method will compile without error?

- (A) `Sample()`
- (B) `Sample(int x = 10, double y = 6.2)`
- (C) `Sample(int x, double y)`
- (D) `Sample(10, 6.2)`
- (E) `Sample(6.2, 10)`

unit5_final_review

5. Consider the following class declarations.

```
public class Point
{
    private double x;    // x-coordinate
    private double y;    // y-coordinate

    public Point()
    {
        x = 0;
        y = 0;
    }

    public Point(double a, double b)
    {
        x = a;
        y = b;
    }

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

public class Circle
{
    private Point center;
    private double radius;

    /** Constructs a circle where (a, b) is the center and r is the radius.
     */
    public Circle(double a, double b, double r)
    {
        /* missing code */
    }
}
```

Which of the following replacements for `/* missing code */` will correctly implement the `Circle` constructor?

unit5_final_review

- I. `center = new Point();`
`radius = r;`
- II. `center = new Point(a, b);`
`radius = r;`
- III. `center = new Point();`
`center.x = a;`
`center.y = b;`
`radius = r;`

- (A) I only
- (B) II only
- (C) III only
- (D) II and III only
- (E) I, II, and III

unit5_final_review

6. 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 */*.

- I. Double d1 = new Double(temperature); Double d2 = new Double(other.temperature);
return d1.compareTo(d2);
- II. if (temperature < other.temperature)

 return -1;

else if (temperature == other.temperature)

unit5_final_review

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?

- (A) II only
- (B) I and II only
- (C) I and III only
- (D) II and III only
- (E) I, II, and III

7. Consider the following methods.

```
public void changer(String x, int y)
{
    x = x + "peace";
    y = y * 2;
}

public void test()
{
    String s = "world";
    int n = 6;
    changer(s, n);

    /* End of method */
}
```

When the call `test ()` is executed, what are the values of `s` and `n` at the point indicated by */* End of method */* ?

unit5_final_review

- (A) s / n
world / 6
- (B) s / n
worldpeace / 6
- (C) s / n
world / 12
- (D) s / n
worldpeace / 12
- (E) s / n
peace / 12

8.

Consider the following method.

```
public void mystery(int[] data)
{
    for (int k = 0; k < data.length - 1; k++)
        data[k + 1] = data[k] + data[k + 1];
}
```

The following code segment appears in another method in the same class.

```
int[] values = {5, 2, 1, 3, 8};
mystery(values);
for (int v : values)
    System.out.print(v + " ");
System.out.println();
```

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

- (A) 5 2 1 3 8
- (B) 5 7 3 4 11
- (C) 5 7 8 11 19
- (D) 7 3 4 11 8
- (E) Nothing is printed because an `ArrayIndexOutOfBoundsException` is thrown during the execution of method `mystery`.

unit5_final_review

9. Consider the following two methods, which appear within a single class.

```
public static void changeIt(int[] arr, int val, String word)
{
    arr = new int[5];
    val = 0;
    word = word.substring(0, 5);

    for (int k = 0; k < arr.length; k++)
    {
        arr[k] = 0;
    }
}

public static void start()
{
    int[] nums = {1, 2, 3, 4, 5};
    int value = 6;
    String name = "blackboard";

    changeIt(nums, value, name);

    for (int k = 0; k < nums.length; k++)
    {
        System.out.print(nums[k] + " ");
    }

    System.out.print(value + " ");
    System.out.print(name);
}
```

What is printed as a result of the call `start()` ?

- (A) 0 0 0 0 0 0 black
- (B) 0 0 0 0 0 6 blackboard
- (C) 1 2 3 4 5 6 black
- (D) 1 2 3 4 5 0 black
- (E) 1 2 3 4 5 6 blackboard

unit5_final_review

10. Consider the following class definition.

```
public class Example
{
    private int x;
    // Constructor not shown.
}
```

Which of the following is a correct header for a method of the `Example` class that would return the value of the private instance variable `x` so that it can be used in a class other than `Example` ?

- (A) `private int getX()`
- (B) `private void getX()`
- (C) `public int getX()`
- (D) `public void getX()`
- (E) `public void getX(int x)`

unit5_final_review

11. Consider the following class declaration.

```
public class IntCell
{
    private int myStoredValue;

    // constructor not shown

    public int getValue()
    {
        return myStoredValue;
    }

    public String toString ()
    {
        return "" + myStoredValue;
    }
}
```

Assume that the following declaration appears in a client class.

```
IntCell m = new IntCell();
```

Which of these statements can be used in the client class?

- I. `System.out.println(m.getValue());`
 - II. `System.out.println(m.myStoredValue);`
 - III. `System.out.println(m);`
- (A) I only
 - (B) II only
 - (C) III only
 - (D) I and II
 - (E) I and III

unit5_final_review

12. Consider the following instance variables and method that appear in a class representing student information.

```
private int assignmentsCompleted;
```

```
private double testAverage;
```

```
public boolean isPassing()
```

```
{ /* implementation not shown */ }
```

A student can pass a programming course if at least one of the following conditions is met.

- The student has a test average that is greater than or equal to 90.
- The student has a test average that is greater than or equal to 75 and has at least 4 completed assignments.

Consider the following proposed implementations of the `isPassing` method.

- I. `if (testAverage >= 90)`

```
    return true;
```

```
    if (testAverage >= 75 && assignmentsCompleted >= 4)
```

```
        return true;
```

```
    return false;
```

- II. `boolean pass = false;`

```
    if (testAverage >= 90)
```

```
        pass = true;
```

```
    if (testAverage >= 75 && assignmentsCompleted >= 4)
```

```
        pass = true;
```

```
    return pass;
```

- III. `return (testAverage >= 90) ||`

unit5_final_review

(testAverage >= 75 && assignmentsCompleted >= 4);

Which of the implementations will correctly implement method isPassing?

- (A) I only
 - (B) II only
 - (C) I and III only
 - (D) II and III only
 - (E) I, II, and III
-

Directions: Select the choice that best fits each statement. The following question(s) refer to the following information.

Consider the following partial class declaration.

```
public class SomeClass
{
    private int myA;
    private int myB;
    private int myC;

    // Constructor(s) not shown

    public int getA()
    { return myA; }

    public void setB(int value)
    { myB = value; }
}
```

13. Which of the following changes to SomeClass will allow other classes to access but not modify the value of myC ?

unit5_final_review

- (A) Make myC public.

Include the method:

- (B)

```
public int getC()  
{ return myC; }
```

Include the method:

- (C)

```
private int getC()  
{ return myC; }
```

Include the method:

- (D)

```
public void getC(int x)  
{ x = myC; }
```

Include the method:

- (E)

```
private void getC(int x)  
{ x = myC; }
```
-

unit5_final_review

14. Consider the following instance variables and incomplete method that are part of a class that represents an item. The variables `years` and `months` are used to represent the age of the item, and the value for `months` is always between 0 and 11, inclusive. Method `updateAge` is used to update these variables based on the parameter `extraMonths` that represents the number of months to be added to the age.

```
private int years;

private int months; // 0 <= months <= 11

// precondition: extraMonths >= 0

public void updateAge(int extraMonths)
{
    /* body of updateAge */
}
```

Which of the following code segments could be used to replace */* body of updateAge */* so that the method will work as intended?

- I. `int yrs = extraMonths % 12;`
`int mos = extraMonths / 12;`
`years = years + yrs;`
`months = months + mos;`
- II. `int totalMonths = years * 12 + months + extraMonths;`
`years = totalMonths / 12;`
`months = totalMonths % 12;`
- III. `int totalMonths = months + extraMonths;`
`years = years + totalMonths / 12;`
`months = totalMonths % 12;`

unit5_final_review

- (A) I only
- (B) II only
- (C) III only
- (D) II and III only
- (E) I, II, and III