U5-10_1

Name

1. A car dealership needs a program to store information about the cars for sale. For each car, they want to keep track of the following information: number of doors (2 or 4), whether the car has air conditioning, and its average number of miles per gallon. Which of the following is the best object-oriented program design?

(A) Use one class, Car, with three instance variables: int numDoors, boolean hasAir, and double milesPerGallon.

(B) Use four unrelated classes: Car, Doors, AirConditioning, and MilesPerGallon.

(C) Use a class Car with three subclasses: Doors, AirConditioning, and MilesPerGallon.

(D) Use a class Car, with a subclass Doors, with a subclass AirConditioning, with a subclass MilesPerGallon.

(E) Use three classes: Doors, AirConditioning, and MilesPerGallon, each with a subclass Car.

U5-10_1

2. 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?

U5-10_1

(A) I only

(B) II only

(C) III only

(D) I and II only

(E) I, II, and III

---

3. The Car class will contain two string attributes for a car's make and model. The class will also contain a constructor.

```
public class Car
{
/* missing code */
}
```

Which of the following replacements for /* missing code */ is the most appropriate implementation of the class?

U5-10_1

(A)
```
public String make;
public String model;
public Car(String myMake, String myModel)
{ /* implementation not shown */ }
```

(B)
```
public String make;
public String model;
private Car(String myMake, String myModel)
{ /* implementation not shown */ }
```

(C)
```
private String make;
private String model;
public Car(String myMake, String myModel)
{ /* implementation not shown */ }
```

(D)
```
public String make;
private String model;
private Car(String myMake, String myModel)
( /* implementation not shown */ }
```

(E)
```
private String make;
private String model;
private Car(String myMake, String myModel)
{ /* implementation not shown */ }
```

---

4. The Date class below will contain three int attributes for day, month, and year, a constructor, and a setDate method. The setDate method is intended to be accessed outside the class.

```
public class Date
{
/* missing code */
}
```

Which of the following replacements for /* missing code */ is the most appropriate implementation of the class?

U5-10_1

```
private int day;
private int month;
private int year;
private Date()
{ /* implementation not shown */ }
private void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

(A)

```
private int day;
private int month;
private int year;
public Date()
{ /* implementation not shown */ }
private void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

(B)

```
private int day;
private int month;
private int year;
public Date()
{ /* implementation not shown */ }
public void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

(C)

```
public int day;
public int month;
public int year;
private Date()
{ /* implementation not shown */ }
private void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

(D)

```
public int day;
public int month;
public int year;
public Date()
{ /* implementation not shown */ }
public void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

(E)

U5-10_1

5. The Player class below will contain two int attributes and a constructor. The class will also contain a method getScore that can be accessed from outside the class.

```
public class Player
{
/* missing code */
}
```

Which of the following replacements for /* missing code */ is the most appropriate implementation of the class?

U5-10_1

(A)
```
private int score;
private int id;
private Player(int playerScore, int playerID)
{ /* implementation not shown */ }
private int getScore()
{ /* implementation not shown */ }
```

(B)
```
private int score;
private int id;
public Player(int playerScore, int playerID)
{ /* implementation not shown */ }
private int getScore()
{ /* implementation not shown */ }
```

(C)
```
private int score;
private int id;
public Player(int playerScore, int playerID)
{ /* implementation not shown */ }
public int getScore()
{ /* implementation not shown */ }
```

(D)
```
public int score;
public int id;
public Player(int playerScore, int playerID)
{ /* implementation not shown */ }
private int getScore()
{ /* implementation not shown */ }
```

(E)
```
public int score;
public int id;
public Player(int playerScore, int playerID)
{ /* implementation not shown */ }
public int getScore()
{ /* implementation not shown */ }
```

U5-10_1

6. Consider the following code segment.

```
ArrayList<String> animals = new ArrayList<>();
animals.add("fox");
animals.add(0, "squirrel");
animals.add("deer");
animals.set(2, "groundhog");
animals.add(1, "mouse");
System.out.println(animals.get(2) + " and " + animals.get(3));
```

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

(A) `mouse and fox`

(B) `fox and groundhog`

(C) `groundhog and deer`

(D) `fox and deer`

(E) `squirrel and groundhog`

---

7. Consider the following code segment.

```
ArrayList<Integer> oldList = new ArrayList();
oldList.add(100);
oldList.add(200);
oldList.add(300);
oldList.add(400);
ArrayList<Integer> newList = new ArrayList();
newList.add(oldList.remove(1));
newList.add(oldList.get(2));
System.out.println(newList);
```

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

U5-10_1

(A) [100, 300, 400]

(B) [200, 300]

(C) [200, 400]

(D) Nothing is printed because the code segment does not compile.

(E) Nothing is printed because an IndexOutOfBoundsException will occur.

---

8. Consider the following code segment.

```
ArrayList<Double> conditionRating = new ArrayList<Double>();
conditionRating.add(9.84);
conditionRating.add(8.93);
conditionRating.add(7.65);
conditionRating.add(6.24);
conditionRating.remove(2);
conditionRating.set(2, 7.63);
System.out.println(conditionRating);
```

What is printed when this code segment is executed?

(A) `[9.84, 7.63, 6.24]`

(B) `[9.84, 7.63, 7.65, 6.24]`

(C) `[9.84, 8.93, 7.63]`

(D) `[9.84, 8.93, 7.63, 6.24]`

(E) `[9.84, 8.93, 7.65, 7.63]`

U5-10_1

9. Consider the following method.

```
public static void addOneToEverything(int[] numbers)
{
for (int j = 0; j < numbers.length; j++)
{
numbers[j]++;
}
}
```

Which of the following code segments, if any, can be used to replace the body of the method so that numbers will contain the same values?

I.

```
for (int num : numbers)
{
num++;
}
```

II.

```
for (int num : numbers)
{
num[j]++;
}
```

III.

```
for (int num : numbers)
{
numbers[num]++;
}
```

(A) I only

(B) I and III only

(C) II and III only

(D) I, II, and III

(E) None of the code segments will return an equivalent result.

U5-10_1

10. Consider the following class definitions.

```
public class Robot
{
private int servoCount;
public int getServoCount()
{
return servoCount;
}
public void setServoCount(int in)
{
servoCount = in;
}
}
public class Android extends Robot
{
private int servoCount;
public Android(int initVal)
{
setServoCount(initVal);
}
public int getServoCount()
{
return super.getServoCount();
}
public int getLocal()
{
return servoCount;
}
public void setServoCount(int in)
{
super.setServoCount(in);
}
public void setLocal(int in)
{
servoCount = in;
}
}
```

The following code segment appears in a method in another class.

```
int x = 10;
int y = 20;
/* missing code */
```

Which of the following code segments can be used to replace /* missing code */ so that the value 20 will be printed?

U5-10_1

(A)
```
Android a = new Android(x);
a.setServoCount(y);
System.out.println(a.getServoCount());
```

(B)
```
Android a = new Android(x);
a.setServoCount(y);
System.out.println(a.getLocal());
```

(C)
```
Android a = new Android(x);
a.setLocal(y);
System.out.println(a.getServoCount());
```

(D)
```
Android a = new Android(y);
a.setServoCount(x);
System.out.println(a.getLocal());
```

(E)
```
Android a = new Android(y);
a.setLocal(x);
System.out.println(a.getLocal());
```

---

11. Consider the following statement, which is intended to create an ArrayList named theater_club to store elements of type Student. Assume that the Student class has been properly defined and includes a no-parameter constructor.

```
ArrayList<Student> theater_club = new /* missing code */;
```

Which choice can replace /* missing code */ so that the statement compiles without error?

(A) `Student()`

(B) `Student ArrayList()`

(C) `ArrayList(Student)`

(D) `ArrayList<Student>()`

(E) `ArrayList<theater_club>()`

U5-10_1

12. Consider the following class definitions.

```
public class Artifact
{
private String title;
private int year;
public Artifact(String t, int y)
{
title = t;
year = y;
}
public void printInfo()
{
System.out.print(title + " (" + year + ")");
}
}
public class Artwork extends Artifact
{
private String artist;
public Artwork(String t, int y, String a)
{
super(t, y);
artist = a;
}
public void printInfo()
{
/* missing implementation */
}
}
```

The following code segment appears in a method in another class.

```
Artwork starry = new Artwork("The Starry Night", 1889, "Van Gogh");
starry.printInfo();
```

The code segment is intended to produce the following output.

```
The Starry Night (1889) by Van Gogh
```

Which of the following can be used to replace /* missing implementation */ in the printInfo method in the Artwork class so that the code segment produces the intended output?

U5-10_1

(A)
```
System.out.print(title + " (" + year + ") by " + artist);
```

(B)
```
super.printInfo(artist);
```

(C)
```
System.out.print(super.printInfo() + " by " + artist);
```

(D)
```
super();
System.out.print(" by " + artist);
```

(E)
```
super.printInfo();
System.out.print(" by " + artist);
```

---

13. Assume that myList is an ArrayList that has been correctly constructed and populated with objects. Which of the following expressions produces a valid random index for myList?

(A) (int) (Math.random () * myList.size ()) - 1

(B) (int) (Math.random () * myList.size ())

(C) (int) (Math.random () * myList.size ()) + 1

(D) (int) (Math.random () * (myList.size () + 1))

(E) Math.random (myList.size ())

---

14. Assume that the array arr has been defined and initialized as follows.

```
int[] arr = /* initial values for the array */ ;
```

Which of the following will correctly print all of the odd integers contained in arr but none of the even integers contained in arr ?

U5-10_1

(A)
```
for (int x : arr)
  if (x % 2 != 0)
    System.out.println(x);
```

(B)
```
for (int k = 1; k < arr.length; k++)
  if (arr[k] % 2 != 0)
    System.out.println(arr[k]);
```

(C)
```
for (int x : arr)
  if (x % 2 != 0)
    System.out.println(arr[x]);
```

(D)
```
for (int k = 0; k < arr.length; k++)
  if (arr[k] % 2 != 0)
    System.out.println(k);
```

(E)
```
for (int x : arr)
  if (arr[x] % 2 != 0)
    System.out.println(arr[x]);
```

15. Consider the following class definition.

```
public class Backyard
{
private int length;
private int width;
public Backyard(int l, int w)
{
length = l;
width = w;
}
public int getLength()
{
return length;
}
public int getWidth()
{
return width;
}
public boolean equals(Object other)
{
if (other == null)
{
return false;
}
Backyard b = (Backyard) object;
return (length == b.getLength() && width == b.getWidth());
}
}
```

The following code segment appears in a class other than Backyard. It is intended to print true if b1 and b2 have the same lengths and widths, and to print false otherwise. Assume that x, y, j, and k are properly declared and initialized variables of type int.

```
Backyard b1 = new Backyard(x, y);
Backyard b2 = new Backyard(j, k);
System.out.println( /* missing code */ );
```

Which of the following can be used as a replacement for /* missing code */ so the code segment works as intended?

U5-10_1

(A) `b1 == b2`

(B) `b1.equals(b2)`

(C) `equals(b1, b2)`

(D) `b1.equals(b2.getLength(), b2.getWidth())`

(E) `b1.length == b2.length && b1.width == b2.width`

16. Consider the following class definition.

```
public class Element
{
public static int max_value = 0;
private int value;
public Element (int v)
{
value = v;
if (value > max_value)
{
max_value = value;
}
}
}
```

The following code segment appears in a class other than Element.

```
for (int i = 0; i < 5; i++)
{
int k = (int) (Math.random() * 10 + 1);
if (k >= Element.max_value)
{
Element e = new Element(k);
}
}
```

Which of the following best describes the behavior of the code segment?

U5-10_1

(A) Exactly 5 Element objects are created.

(B) Exactly 10 Element objects are created.

(C) Between 0 and 5 Element objects are created, and Element.max_value is increased only for the first object created.

(D) Between 1 and 5 Element objects are created, and Element.max_value is increased for every object created.

(E) Between 1 and 5 Element objects are created, and Element.max_value is increased for at least one object created.

---

17. Consider the following class definition.

```
public class WordClass
{
private final String word;
private static String max_word = "";
public WordClass (String s)
{
word = s;
if (word.length() > max_word.length())
{
max_word = word;
}
}
}
```

Which of the following is a true statement about the behavior of WordClass objects?

U5-10_1

(A) A WordClass object can change the value of the variable word more than once.

(B) Every time a WordClass object is created, the max_word variable is referenced.

(C) Every time a WordClass object is created, the value of the max_word variable changes.

(D) No two WordClass objects can have their word length equal to the length of max_word.

(E) The value of the max_word variable cannot be changed once it has been initialized.

U5-10_1

18. Consider the following class definition.

```
public class Beverage
{
private int temperature;
public Beverage(int t)
{
temperature = t;
}
public int getTemperature()
{
return temperature;
}
public boolean equals(Object other)
{
if (other == null)
{
return false;
}
Beverage b = (Beverage) other;
return (b.getTemperature() == temperature);
}
}
```

The following code segment appears in a class other than Beverage. Assume that x and y are properly declared and initialized int variables.

```
Beverage hotChocolate = new Beverage(x);
Beverage coffee = new Beverage(y);
boolean same = /* missing code */;
```

Which of the following can be used as a replacement for /* missing code */ so that the boolean variable same is set to true if and only if the hotChocolate and coffee objects have the same temperature values?

U5-10_1

A `(hotChocolate = coffee)`

B `(hotChocolate == coffee)`

C `hotChocolate.equals(coffee)`

D `hotChocolate.equals(coffee.getTemperature())`

E `hotChocolate.getTemperature().equals(coffee.getTemperature())`

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

Consider the following sort method. This method correctly sorts the elements of array data into increasing order.

```
public static void sort(int[] data)
{
  for (int j = 0; j < data.length - 1; j++)
  {
    int m = j;
    for (int k = j + 1; k < data.length; k++)
    {
      if (data[k] < data[m])      /* Compare values */
      {
        m = k;
      }
    }
    int temp = data[m];           /* Assign to temp */
    data[m] = data[j];
    data[j] = temp;

    /* End of outer loop */
  }
}
```

U5-10_1

19. Assume that sort is called with the array {1, 2, 3, 4, 5, 6}. How many times will the expression indicated by /* *Compare values* */ and the statement indicated by /* *Assign to temp* */ execute?

(A) Compare values / Assign to temp
15 / 0

(B) Compare values / Assign to temp
15 / 5

(C) Compare values / Assign to temp
15 / 6

(D) Compare values / Assign to temp
21 / 5

(E) Compare values / Assign to temp
21 / 6

20. Assume that sort is called with the array {6, 3, 2, 5, 4, 1}. What will the value of data be after three passes of the outer loop (i.e., when j = 2 at the point indicated by / * *End of outer loop* * /) ?

(A) {1, 2, 3, 4, 5, 6}

(B) {1, 2, 3, 5, 4, 6}

(C) {1, 2, 3, 6, 5, 4}

(D) {1, 3, 2, 4, 5, 6}

(E) {1, 3, 2, 5, 4, 6}