Name: CAELAN KIMBALL **ID: 020022**
Date: 05/23/2019 **Class: Computer Science 2 2018**
Score: 106 / 108 (98.15%)

2018-2019 Computer Science 2

TRUE/FALSE

T 1. A data structure is a data type whose components are smaller data structures and/or simple data types.

Points: 1 / 1

T 2. A record is a data structure with one, or more, elements, called fields, of the same or different data types.

Points: 1 / 1

T 3. A file is an external data structure with a specified number of elements assigned to an internal file name.

Points: 1 / 1

T 4. The file data structure allows transfer of data between internal and external storage.

Points: 1 / 1

T 5. The **Set** class is abstract and can not be used to construct an object directly.

Points: 1 / 1

T 6. A method can be used as an argument for another method.

Points: 1 / 1

T 7. To use a programming "black box" (subroutine), you need to know about its interface.

Points: 1 / 1

T 8. Subroutines in Java can be either static or non-static.

Points: 1 / 1

T 9. Every subroutine in Java must be defined inside a class.

Points: 1 / 1

T 10. It is not legal to have one subroutine physically nested inside another.

Points: 1 / 1

MULTIPLE CHOICE

A 11. Which of the following statements, if added to the code segment below, would output the length of *a*?

```
char a[] = {'a','b','c','d','e'};
String s = "";
for (int i = 0; i < 5; i++)
s += a[i];
System.out.println(s.substring(0,3));
```

a. out.print(a.length);
b. out.print(a.length());
c. out.print(a.size());
d. more than one of the above

Points: 1 / 1

B 12. Which of the following will print the number of elements in an array *a*?

a. System.out.print(a.size());
b. System.out.print(a.length);
c. System.out.print(a.length());

Points: 1 / 1

B 13. What is the output of the code segment at right?

```
try{
      int[] array1 = {1, 2, 3};
      int[] array2 = {4, 3, 2, 1};
      for(int i=0; i<array2.length; i++){
            System.out.print(array2[i]);
            array1[i] = array2[i];
      }
}catch(Exception e){
      System.out.println("FAIL");
}
```

a. FAIL
b. 4321FAIL
c. 432FAIL
d. 4321
e. 432

Points: 1 / 1

A 14. What is the output of the code segment at right?

```
List<String>list = new
ArrayList<String>( );
list.add("dog");
list.add("cat");
int sum = 0;
for (String word : list){
if (word.indexOf ('m') > 6)
sum ++;
}
System.out.print(sum);
```

a. 0
b. 1
c. 2
d. error
e. none of the above

Points: 1 / 1

D 15. A data structure is a data type

a. with a single value.
b. with two or more values.
c. with one or more simple data types.
d. whose components are smaller data structures and/or simple data types.

Points: 1 / 1

C 16. Data structures are defined by

a. the data types they store only.
b. the manner of data accesses only.
c. both the data storage and the data access.
d. the storage of primitive data types.

Points: 1 / 1

C 17. Consider the two program segments below.

Segment1	Segment2
int list[ ]; **list = new int[100];**	**int list[ ] = new int[100];**

Which of the following is a true statement about the comparison of Segment1 and Segment2?

a. Segment1 declares **list** correctly. Segment2 declares **list** incorrectly.
b. Segment1 declares **list** incorrectly. Segment2 declares **list** correctly.
c. Both Segment1 and Segment2 declare **list** correctly.
d. Both Segment1 and Segment2 declare **list** incorrectly.

Points: 1 / 1

D 18. What is the output of program **Java1212.java** below?

```
public class Java1212
{
  public static void main(String args[ ])
  {
    int list[ ];
    list = new int[10];
    for (int k = 0; k < 10; k++)
      System.out.print(list[k] + " ");
    System.out.println();
  }
}
```

a. `0 1 2 3 4 5 6 7 8 9`
b. `1 2 3 4 5 6 7 8 9 10`
c. `0 0 0 0 0 0 0 0 0 0`
d. `0 0 0 0 0 0 0 0 0 0`

Points: 1 / 1

B 19. What is the output of program **Java1213.java** below?

```
public class Java1213
{
  public static void main(String args[ ])
  {
    char list[ ] = new char[5];
    System.out.print("Boo");
    for (int k = 0; k < list.length; k++)
      System.out.print(list[k]);
    System.out.println("hiss");
  }
}
```

a. `Boohiss`
b. `Boo  hiss`
c. `Boo00000hiss`
d. `Boo`

`hiss`

Points: 1 / 1

B 20. What is the FIRST and LAST output from this program segment?

```
int IntNum[] = new int[100];
 int J;
for (J=0; J<100; J++)
        IntNum[J] = J;
for (J=0; J<100; J++)
        System.out.println(IntNum[J]);
```

a. 0 and 100
b. 0 and 99
c. 1 and 100
d. 1 and 99
e. Array Index Out Of Bounds Error

Points: 1 / 1

D 21. **Use this program segment to answer the question.**

```
boolean George[] = new boolean[15];
int J;

System.out.println(George.length);

for (J=0; J<15; J++)
        if (J == 0)
                George [J] = (J==0);
        else
                George [J] = !George[J-1];

System.out.println(George[7]);

System.out.println(George[8]);

System.out.println(George[15]);
```

What is the output of the first **println**?
a. true
b. false
c. 14
d. 15
e. Array Index Out Of Bounds Error

Points: 1 / 1

B 22. **Use this program segment to answer the question.**

```
boolean George[] = new boolean[15];
int J;

System.out.println(George.length);

for (J=0; J<15; J++)
        if (J == 0)
                George [J] = (J==0);
        else
                George [J] = !George[J-1];

System.out.println(George[7]);

System.out.println(George[8]);

System.out.println(George[15]);
```

What is the output of the second **println**?

a. true
b. false
c. 14
d. 15
e. Array Index Out Of Bounds Error

Points: 1 / 1

A 23. **Use this program segment to answer the question.**

```
boolean George[] = new boolean[15];
int J;

System.out.println(George.length);

for (J=0; J<15; J++)
        if (J == 0)
                George [J] = (J==0);
        else
                George [J] = !George[J-1];

System.out.println(George[7]);

System.out.println(George[8]);

System.out.println(George[15]);
```

What is the output of the third **println**?

a. true
b. false
c. 14
d. 15
e. Array Index Out Of Bounds Error

Points: 1 / 1

E 24. **Use this program segment to answer the question.**

```
boolean George[] = new boolean[15];
int J;

System.out.println(George.length);

for (J=0; J<15; J++)
        if (J == 0)
                George [J] = (J==0);
        else
                George [J] = !George[J-1];

System.out.println(George[7]);

System.out.println(George[8]);

System.out.println(George[15]);
```

What is the output of the fourth **println**?

a. true
b. false
c. 14
d. 15
e. Array Index Out Of Bounds Error

Points: 1 / 1

B 25. Which of the following statement displays the **list** elements correctly?

int list[ ] = {11,22,33,44,55,66,77,88,99};

a.
```
for (int k=0; list item; k++)
        System.out.print(item + "  ");
```
b.
```
for (int item: list)
        System.out.print(item + "  ");
```
c.
```
for (int k=0; int item; k++)
        System.out.print(item + "  ");
```
d.
```
for (int k=0; list item; k++)
        System.out.print(item[k] + "  ");
```

Points: 1 / 1

C 26. The **Arrays** class

a. makes it possible to display individual array elements using the new Java 5.0 loop.
b. makes it possible to display individual array elements with any type of loop control structure.
c. makes it possible to display individual array elements without using any type of control structure.
d. does not make it possible to display individual array elements.

Points: 1 / 1

E 27. Which of the following are **Arrays** class methods?

a. **toString**
b. **fill**
c. **binarySearch**
d. **sort**
e. All of the above

Points: 1 / 1

A 28. An array is a

a. data structure with one, or more, elements of the same type.
b. data structure with LIFO access.
c. data structure, which allows transfer between internal and external storage.
d. data structure with one, or more, elements, called fields, of the same or different data types.

Points: 1 / 1

C 29. Consider the **mambo** object declaration below.

```
double mambo[ ][ ];
mambo = new double[4][5];
int r;  // row index on mambo
int c;  // column index of mambo
```

Which of the following statements stores the *column length* of **mambo**?

a. **mambo.length**
b. **mambo.rowLength**
c. **mambo[r].length**
d. **mambo[c].length**

Points: 1 / 1

C 30. Since built-in Java array is already a class, is there any reason to declare a **List** class, which contains an integer, character or other type of array?

a. No. Arrays function fine without any additional class declarations.
b. It is strictly personal preference. Creating another class adds readability to the program.
c. Yes. The built-in array does not provide any methods to process array elements.
d. Yes. Placing the built-in array inside a class increases execution efficiency.

Points: 1 / 1

A 31. Use the **List** class below for the questions.

```
class List
{
        private int intArray[];
        private int size;
        public List(int s)
        {
                size = s;
                intArray = new int[size];
        }

        public void display()
        {
                for (int k = 0; k < size; k++)
                        System.out.print(intArray[k] + "  ");
                System.out.println();
        }
}
```

Assume that **qwerty** is an object of the **List** class. Which of the following statements accesses members of the **List** class correctly?

a. **qwerty.display();**
b. **int k = 5;**
   qwerty.intArray[k] = 100;
c. **qwerty.intArray[] = new List[100];**
d. **qwerty.size = 100;**

Points: 1 / 1

A 32. An array of 1500 ordered elements requires at most _____ comparisons to find a search item with a binary search.

a. 11
b. 12
c. 750
d. 1500

Points: 1 / 1

D 33. Which of the following code segments correctly defines list as an array data structure with the potential to store 1000 elements of any type?

I. **ArrayList list = new ArrayList();**
II. **ArrayList[ ] list = new ArrayList[1000];**
III. **ArrayList list = new ArrayList(1000);**

a. II only
b. I and II only
c. II and III only
d. I, II and III

Points: 1 / 1

A 34. Consider the following code segment.

```
ArrayList names = new ArrayList();
names.add("John");
names.add("Greg");
names.add("Maria");
names.add("Heidi");
names.remove(1);
names.remove(2);
System.out.println();
for (int k = 0; k < names.size(); k++)
        System.out.print(names.get(k) + "  ");
```

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

a. John Maria
b. John Heidi
c. Greg Heidi
d. Greg Maria

Points: 1 / 1

A 35. Rewrite the old **for** loop program segment below with the new **for** loop.

```
int list[ ] = {1,2,3,4,5,6};
for (int k = 0; k < list.length; k++)
        System.out.println(list[k]);
```

a. for (int number: list)
   System.out.print(number + " ");
b. for (int number: list.length)
   System.out.print(number + " ");
c. for (int k = 0; number: list)
   System.out.print(number[k]);
d. This program segment cannot be converted to the new for loop.

Points: 1 / 1

D 36. Rewrite the old **for** loop program segment below with the new **for** loop.

```
for (int k = 0; k < 10; k++)
        System.out.println(k);
```

a. for (int number: k)
   System.out.print(number + " ");
b. for (int number: k.length)
   System.out.print(k + " ");
c. for (int k = 0; number: list)
   System.out.print(number[k]);
d. This program segment cannot be converted to the new for loop.

Points: 1 / 1

A 37. What is the index of the first element in an ArrayList?

a. 0
b. 10
c. 1
d. It can not be determined.

Points: 1 / 1

B 38. What is the index of the last element in an ArrayList called stuff?

a. stuff.lastIndex()
b. stuff.size()
c. stuff.legth
d. It can not be determined.

Points: 1 / 1

A 39. What is the output of this program?

```
import java.util.ArrayList;

public class Java2014
{
    public static void main(String args[])
    {
        ArrayList names = new ArrayList();
        names.add("Isolde");
        names.add("John");
        names.add("Greg");
        names.add("Maria");
        names.add(new String("Heidi"));

        System.out.println("names contains " + names);
        System.out.println();
    }
}
```

a. `names contains [Isolde, John, Greg, Maria, Heidi]`
b. `names contains [Heidi, Maria, Greg, John, Isolde]`
c. `names contains [John, Greg, Maria, Heidi]`
d. `names contains [Isolde, John, Greg, Maria]`
e. Error

Points: 1 / 1

D 40. Which of the following code segments correctly adds a new element to the list array?

Code Segment 1
```
ArrayList array = new ArrayList();
array.add("9999");
```

Code Segment 2
```
ArrayList array = new ArrayList();
array.add(new Integer(9999));
```

Code Segment 3
```
ArrayList array = new ArrayList();
array.add(new String("Aardvark"));
```

a. Code Segment 1 only
b. Code Segments 1 and 2 only
c. Code Segments 2 and 3 only
d. Code Segments 1, 2 and 3

Points: 1 / 1

D 41. What is the output of this program?

```
import java.util.ArrayList;

public class Java1322
{
    public static void main(String args[])
    {
        ArrayList names = new ArrayList();
        names.add("Isolde");
        names.add("John");
        names.add("Greg");
        names.add("Maria");
        names.add("Heidi");
        System.out.println(names.contains("Greg"));
    }
}
```

a. -1
b. 2
c. 3
d. true
e. false

Points: 1 / 1

E 42. What is the output of this program?

```
import java.util.ArrayList;

public class Java1323
{
    public static void main(String args[])
    {
        ArrayList names = new ArrayList();
        names.add("Isolde");
        names.add("John");
        names.add("Greg");
        names.add("Maria");
        names.add("Heidi");
        System.out.println(names.contains("Jessica"));
    }
}
```

a. -1
b. 2
c. 3
d. true
e. false

Points: 1 / 1

A 43. What is the output of this program?

```
import java.util.ArrayList;

public class Java1324
{
    public static void main(String args[])
    {
        ArrayList names = new ArrayList();
        names.add("Isolde");
        names.add("John");
        names.add("Greg");
        names.add("Maria");
        names.add("Heidi");
        System.out.println(names.indexOf("Jessica"));
    }
}
```

a. -1
b. 2
c. 3
d. true
e. false

Points: 1 / 1

B 44. Which of the following interfaces is used to implement the *ArrayList* class?

a. collection
b. list
c. set
d. museum

Points: 1 / 1

A 45. Class methods are typically used when

a. only a single copy of the class needs to be loaded
b. multiple copies or instances of a class are required.
c. it is not necessary to pass information to the methods.
d. only return methods are used in a class.

Points: 1 / 1

C 46. Which of the following statements shows correct syntax to create an object of the **Piggy** class?

a. **Piggy new tom = Piggy();**
b. **Piggy = new tom();**
c. **Piggy tom = new Piggy();**
d. **tom = new Piggy;**

Points: 1 / 1

C 47. Calling an object method requires using

a. a class identifier followed by a dot and a method identifier.
b. a method identifier followed by a dot and a class identifier.
c. an object identifier followed by a dot and a method identifier.
d. a method identifier followed by a dot and an object identifier.

Points: 1 / 1

B 48. When is a constructor called?

a. Each time the constructor identifier is used in a program statement
b. During the instantiation of a new object
c. During the construction of a new class
d. At the beginning of any program execution

Points: 1 / 1

C 49. A class can have

a. one constructor method only.
b. one or two constructor methods.
c. multiple constructors with the same identifier.
d. multiple constructors with different identifiers.

Points: 1 / 1

D 50. Which of the following method declarations can be a constructor?

I.
```
public static void Qwerty()
{
        start = 0;
        max = 1000;
}
```

II.
```
public Qwerty()
{
        start = 0;
        max = 1000;
}
```

III.
```
public Qwerty(int s, int m)
{
        start = s;
        max = m;
}
```

a. I only
b. II only
c. III only
d. II & III only
e. I, II & III

Points: 1 / 1

A 51. Access to **private** data or **private** methods is

a. restricted to methods in the same class.
b. restricted to methods in other classes.
c. available to methods in the same class and other classes.
d. not an issue because the program will not compile.

Points: 1 / 1

A 52. The use of **private** in a class declaration

a. creates greater program reliability.
b. limits data access to authorized program users only.
c. requires password authentication for data access.
d. is required for program compilation.

Points: 1 / 1

E 53. Consider the program below.

```
public class Bacon
{
   public static void main (String args[ ])
   {
      Piggy kathy = new Piggy("Kathy",1500.0);
      Piggy rachel = new Piggy("Rachel",2500.0);
      kathy.showData();                                  // Line 1
      System.out.println("Name    " + rachel.name);     // Line 2
      System.out.println("Savings " + rachel.savings); // Line 3
   }
}

class Piggy
{
   public double savings;
   public String name;

   public Piggy(String n, double s)
   {
      name = n;
      savings = s;
   }

   public void showData()
   {
      System.out.println("Name:    " + name);            // Line 4
      System.out.println("Savings:  " + savings);         // Line 5
   }
}
```

Lines 1 - 5 access data of **Piggy** objects. Which lines have access?

a. Lines 2 and 3 only
b. Lines 4 and 5 only
c. Lines 1, 3, 4 and 5 only
d. Lines 1, 2, 4 and 5 only
e. All five lines have access

Points: 1 / 1

D 54. Consider the program below.

```
public class Bacon
{
   public static void main (String args[ ])
   {
      Piggy kathy = new Piggy("Kathy",1500.0);
      Piggy rachel = new Piggy("Rachel",2500.0);
      kathy.showData();                                  // Line 1
      System.out.println("Name    " + rachel.name);     // Line 2
      System.out.println("Savings " + rachel.savings); // Line 3
   }
}

class Piggy
{
   private double savings;
   public String name;

   public Piggy(String n, double s)
   {
      name = n;
      savings = s;
   }

   public void showData()
   {
      System.out.println("Name:    " + name);            // Line 4
      System.out.println("Savings:  " + savings);        // Line 5
   }
}
```

Lines 1 - 5 access data of **Piggy** objects. Which line(s) have access?

a. Lines 2 and 3 only
b. Lines 4 and 5 only
c. Lines 1, 3, 4 and 5 only
d. Lines 1, 2, 4 and 5 only
e. All five lines have access

Points: 1 / 1

C 55. Consider the program below.

```
public class Bacon
{
   public static void main (String args[ ])
   {
      Piggy kathy = new Piggy("Kathy",1500.0);
      Piggy rachel = new Piggy("Rachel",2500.0);
      kathy.showData();                                  // Line 1
      System.out.println("Name    " + rachel.name);     // Line 2
      System.out.println("Savings " + rachel.savings); // Line 3
   }
}

class Piggy
{
   public double savings;
   private String name;

   public Piggy(String n, double s)
   {
      name = n;
      savings = s;
   }

   public void ShowData()
   {
      System.out.println("Name:    " + name);            // Line 4
      System.out.println("Savings:  " + savings);         // Line 5
   }
}
```

Lines 1 - 5 access data of **Piggy** objects. Which line(s) have access?

a. Lines 2 and 3 only
b. Lines 4 and 5 only
c. Lines 1, 3, 4 and 5 only
d. Lines 1, 2, 4 and 5 only
e. All five lines have access

Points: 1 / 1

C 56. Which class members should be declared as **private**?

a. Data attributes only
b. Methods only
c. Predominantly data attributes and some helper methods
d. Data attributes and constructor methods

Points: 1 / 1

C 57. A **class** method

a. requires using the keyword **new**.
b. requires using the keyword **private**.
c. requires using the keyword **static**.
d. is all of the above.
e. is both A and C.

Points: 1 / 1

A 58. An **object** method

a. requires using the keyword **new**.
b. requires using the keyword **private**.
c. requires using the keyword **static**.
d. is all of the above.
e. is both A and B.

Points: 1 / 1

E 59. A **private** method

I. can only be accessed by methods of the same class.
II. is usually a helper method.
III. can never be a constructor.

a. I only
b. II only
c. III only
d. I & II only
e. I, II & III

Points: 1 / 1

D 60. A **void** method can also be a(n)

a. **static** method.
b. **public** method.
c. **private** method.
d. All of the above
e. A & B only

Points: 1 / 1

A 61. A **default** constructor is a

a. *no-parameter* method, which is called automatically during the instantiation of a new object.
b. *parameter* method, which is called automatically during the instantiation of a new object.
c. *no-parameter* method.
d. *parameter* method.

Points: 1 / 1

B 62. Consider the program below.

```
public class Waco
{
   public static void main (String args[ ])
   {
      Piggy kathy = new Piggy("Kathy",1500.0);
      Piggy rachel = new Piggy("Rachel",2500.0);
      kathy.showData();                                  // Line 1
      System.out.println("Name    " + rachel.name);     // Line 2
      System.out.println("Savings " + rachel.savings); // Line 3
   }
}

class Piggy
{
   private double savings;
   private String name;

   public Piggy(String n, double s)
   {
      name = n;
      savings = s;
   }

   public void showData()
   {
      System.out.println("Name:    " + name);            // Line 4
      System.out.println("Savings:  " + savings);        // Line 5
   }
}
```

Lines 1 - 5 access data of **Piggy** objects. Which lines have access?

a. Lines 4 and 5 only
b. Lines 1, 4 and 5 only
c. Lines 1, 2, 4 and 5 only
d. Lines 1, 3, 4 and 5 only
e. All five lines have access

Points: 1 / 1

C 63. Assume that rand is an object of the Random class.
Which of the following ranges is generated by this statement:
int number = rand.nextInt(250) - 125;

a. [-125..125]

b. [-124..124]

c. [-125..124]

d. [-125..126]

e. [-125..250]

Points: 1 / 1

A 64. What is the output of the following program?

```
import java.util.Random;

public class Question33
{
        public static void main(String args[ ])
        {
                Random rand = new Random();
                rand.setSeed(100);
                System.out.println(rand.nextInt(900) + 100);
                System.out.println(rand.nextInt(900) + 100);
                System.out.println(rand.nextInt(900) + 100);
                System.out.println(rand.nextInt(900) + 100);
                System.out.println(rand.nextInt(900) + 100);
                System.out.println(rand.nextInt(900) + 100);
                System.out.println(rand.nextInt(900) + 100);
                System.out.println(rand.nextInt(900) + 100);
                System.out.println(rand.nextInt(900) + 100);
                System.out.println(rand.nextInt(900) + 100);
        }
}
```

a. 10 different random integers in the **[100..999]** range

b. 10 identical random integers in the **[100..900]** range

c. 10 different random integers in the **[100..900]** range

d. 10 identical random integers in the **[100..1000]** range

Points: 1 / 1

A 65. The kind of output created by an object of the **DecimalFormat** class is determined by

a. the type of parameter used with the construction of a new **DecimalFormat** object.

b. using the **format** method.

c. using the **output** method.

d. all of the above.

Points: 1 / 1

B 66. Complex programs can be broken up into manageable pieces, using _____.

a. black boxes
b. subroutines
c. sledge hammers
d. power saws

Points: 1 / 1

A 67. A subroutine consists of the _____ for carrying out a certain task, grouped together and given a name.

a. instructions
b. steps
c. black box
d. people

Points: 1 / 1

B 68. The code in a subroutine that actually performs the task, how it does what it does, is called the _____.

a. inteface
b. implementation
c. specification
d. documentation

Points: 1 / 1

A 69. The syntactic and semantic specifications of the subroutine.

a. contract
b. statement
c. code
d. GUI

Points: 1 / 1

B 70. The statements between the braces, { and }, in a subroutine definition make up the _____ of the subroutine.

a. head
b. body
c. feet
d. tail

Points: 1 / 1

C 71. Variables that are not part of any subroutine are called _____.

a. local variables
b. return variables
c. member variables
d. default variables

Points: 1 / 1

A 72. Color.RED and Color.YELLOW are public final static variables in the _____ class.

a. Color
b. Font
c. Rainbow
d. Math

Points: 1 / 1

D 73. Non-static members cannot be used in _____ subroutines.

a. hidden
b. missing
c. final
d. static

Points: 1 / 1

D 74. What distinguishes the declaration of a return method?

a. The **return** keyword in the method body
b. The **static** keyword in the method heading
c. a data type declaration in the method heading (do not confuse with parameter data types)
d. Both A and C

Points: 1 / 1

B 75. In the statement **`int num;`** ***int*** is the _____ and ***num*** is the _____.

a. variable identifier data type
b. data type variable identifier
c. class name method name
d. format data type

Points: 1 / 1

A 76. A class is a

a. data structure template or blue print.
b. single instance of a given data structure template
c. collection of primitive data types.
d. a set of statements to perform a specicific task

Points: 1 / 1

C 77. Assume that **rand** is an object of the **Random** class.
Which of the following statements generates a random number in the **[0..1000]** range?

a. **int number = rand.nextInt(1000) + 1;**

b. **int number = rand.nextInt(1000);**

c. **int number = rand.nextInt(1001);**

d. **int number = rand.nextInt(1) + 1000;**

Points: 1 / 1

C 78. Assume that **rand** is an object of the **Random** class.
Which of the following statements generates a random number in the **[41..101]** range?

a. **int number = rand.nextInt(41) + 101;**

b. **int number = rand.nextInt(101) + 41;**

c. **int number = rand.nextInt(61) + 41;**

d. **int number = rand.nextInt(60) + 41;**

Points: 1 / 1

C 79. Access to **public** data or **public** methods is

a. restricted to methods in the same class.
b. restricted to methods in other classes.
c. available to methods in the same class and other classes.
d. not an issue because the program will not compile.

Points: 1 / 1

A 80. Access to **private** data or **private** methods is

a. restricted to methods in the same class.
b. restricted to methods in other classes.
c. available to methods in the same class and other classes.
d. not an issue because the program will not compile.

Points: 1 / 1

C 81. Which features can you use to recognize constructor methods in a class declaration?

I. The constructor identifier is the same as the class identifier.
II. Constructors use both the **public** and the **static** keywords.
III. Constructors are neither void methods nor return methods.

a. I only
b. II only
c. I & III only
d. II & III only
e. I, II & III

Points: 1 / 1

B 82. A **parameterized** constructor is a

a. *no-parameter* method, which is called automatically during the instantiation of a new object.
b. *parameterized* method, which is called automatically during the instantiation of a new object.
c. *no-parameter* method.
d. *parameterized* method.

Points: 1 / 1

C 83. The constructor is used to

a. initialize class data values only.
b. call some initData method.
c. initialize class data and call other methods if they are necessary to construct a new object.
d. only call methods which are necessary to a construct a new object.

Points: 1 / 1

A 84. Which of the following is the minimum class declaration that will compile?

a.
```
class CardDeck
{
}
```

b.
```
class CardDeck
{
        private int numDecks;
}
```

c.
```
class CardDeck
{
        private int numDecks;
        public CardDeck()
        {
                numDecks = 0;
        }
}
```

d.
```
class CardDeck
{
         private int numDecks;
        public CardDeck(int n)
         {
                numDecks = n;
        }
}
```

Points: 1 / 1

D 85. What is an *overloaded* constructor?

a. A constructor with too many program statements.
b. A second constructor with the same constructor heading as the first constructor.
c. A second constructor with a different identifier than the first constructor.
d. A second or other multiple constructor with a different signature than any other constructor.

Points: 1 / 1

B 86. When is a constructor called?

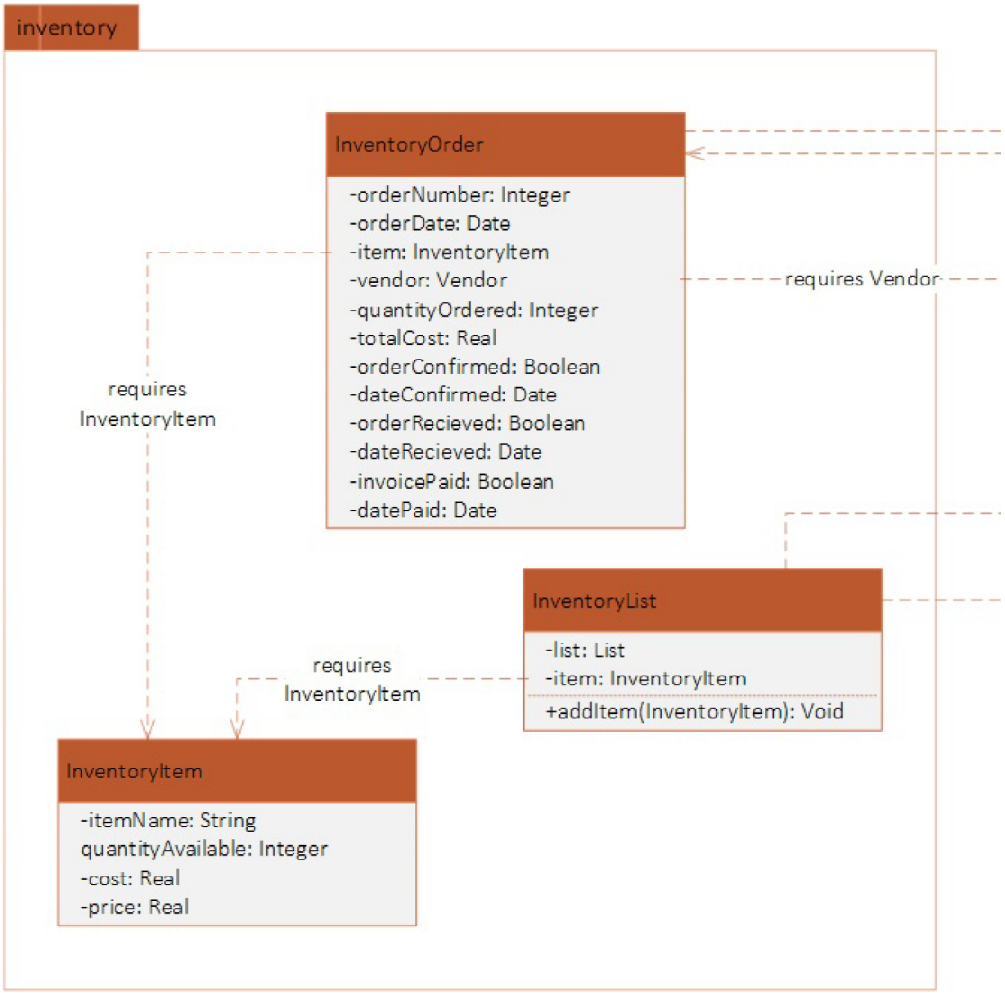
a. Each time the constructor identifier is used in a program statement
b. During the instantiation of a new object
c. During the construction of a new class
d. At the beginning of any program execution

Points: 1 / 1

A 87. An object is

a. one instance of a class.
b. another word for a class.
c. a class with static methods.
d. a method that accesses class attributes.

Points: 1 / 1



A 88. Which of these is correctly based on the UML diagram shown?

a. private String itemName
b. void Date orderDate
c. public String itemName
d. public List list

Points: 1 / 1

D 89. Which of these is correctly based on the UML diagram shown?

a. private String class itemName {
}
b. public class itemName {
}
c. class orderDate {
}
d. class InventoryList {
}

Points: 1 / 1

D 90. Which of these is correctly based on the UML diagram shown?

a. private void addItem(InventoryItem item) {
}
b. public addItem(String s) {
}
c. private addItem(String s) {
}
d. public void addItem(InventoryItem item) {
}

Points: 1 / 1

MATCHING

Match each item with the correct statement below.

a. Application Programming Interface
b. packages
c. Applications Programming
d. importing
e. Import Directive
f. wildcard

C 91. programming using various tools

Points: 1 / 1

A 92. a set of routines, protocols, and tools for building software applications

Points: 1 / 1

B 93. used to organize a group of classes

Points: 1 / 1

E 94. allows utilization of a class without using the full name

Points: 0 / 1

D 95. specific line of code that allows utilization of a class without using the full name

Points: 0 / 1

F 96. the * symbol used to match the name of every class in a package

Points: 1 / 1

Match each item with the correct statement below.

a. default package
b. doc tags
c. static import
d. HTML markup
e. Javadoc
f. jar files

F 97. single Java archive file that can contain many classes

Points: 1 / 1

A 98. the package for classes that are not specifically placed in a package

Points: 1 / 1

E 99. system used to prepare most Java API documentation that can be used to create good API style documentation for any Java class

Points: 1 / 1

D 100. a special code that allows the programmer to use HTML commands

Points: 1 / 1

B 101. commands processed by the Javadoc tool

Points: 1 / 1

C 102. directive that can be used to import static members of a class in the same way that the ordinary import directive imports classes from a package

Points: 1 / 1

Match each item with the correct statement below.

a. postconditions
b. scope
c. named constant
d. hidden
e. contract
f. preconditions

E 103. how a subroutine interacts with the rest of the program

Points: 1 / 1

F 104. a set conditions met before a subroutine is run

Points: 1 / 1

A 105. conditions met after a subroutine is run

Points: 1 / 1

C 106. static member variable that is declared to be final

Points: 1 / 1

B 107. the portion of the program source code where the variable is valid

Points: 1 / 1

D 108. used to describe a member variable that is not visible, due to the scope of a local variable or parameter

Points: 1 / 1