Name: CAELAN KIMBALL **ID: 020022**
Date: 10/15/2018 **Class: Computer Science 2 2018**
Score: 41 / 50 (82%)

CS2 Javanotes Chapter 4 Test 2018

TRUE/FALSE

F 1. A subroutine can not be used inside another subroutine.

Points: 1 / 1

T 2. A method can be used as an argument for another method.

Points: 1 / 1

T 3. To use a programming "black box" (subroutine), you need to know about its interface.

Points: 1 / 1

T 4. Subroutines in Java can be either static or non-static.

Points: 1 / 1

F 5. All of the parameters in a method have to be of the same type?

Points: 1 / 1

MULTIPLE CHOICE

B 6. Complex programs can be broken up into manageable pieces, using _____.

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

Points: 1 / 1

B 7. 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: 0 / 1

A 8. Program subroutines are sometime thought of as_____.

a. black boxes
b. bread boxes
c. impossible to understand
d. TV sets

Points: 1 / 1

A 9. Subroutines in Java are typically called_____.

a. methods
b. programs
c. classes
d. applets

Points: 1 / 1

B 10. The part of a method that we interact with as programmers or users is called the _____.

a. remote
b. interface
c. implementation
d. code

Points: 1 / 1

B 11. 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

B 12. The syntactic and semantic specifications of the subroutine.

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

Points: 0 / 1

B 13. A subroutine definition in Java takes the form:

a.
```
modifiers return-type subroutine-name( ) {
                parameter-list
}
```
b.
```
modifiers return-type subroutine-name( parameter-list ) {
                statements
}
```
c.
```
modifiers parameter-list subroutine-name( return-type ) {
                statements
}
```
d.
```
subroutine-name modifiers return-type( parameter-list ) {
                statements
}
```

Points: 1 / 1

B 14. 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

D 15. Which of the following Java keywords are *access specifiers* used in the declaration of class methods?

a. private
b. public
c. protected
d. All of the above

Points: 1 / 1

A 16. Which of the following method headings uses proper parameter declarations?

a. `public static void guess(double rate, double hours, int deductions)`
b. `public static void guess(double rate, hours, int deductions)`
c. `public static void guess(rate, hours, deductions)`
d. `public static void guess(7.85, 42.5, 3)`

Points: 1 / 1

D 17. Which of the following method calls might use parameters correctly?

a. `guess(double rate, double hours, int deductions);`
b. `guess(double rate, hours, int deductions);`
c. `guess(int rate, hours, deductions);`
d. `guess(7.85, 42.5, 3);`

Points: 1 / 1

A 18. How must class methods be defined?

a. with the static keyword
b. with the non-static keyword
c. with the void keyword
d. with the private keyword

Points: 1 / 1

A 19. How must class methods be accessed?

a. With the class identifier followed by a period and the method identifier
b. With the method identifier followed by a period and the class identifier
c. With the method identifier only
d. With the class identifier followed by a comma and the method identifier

Points: 1 / 1

A 20. All method calls require the use of a set of _____ following the method identifier.

a. parenthesis
b. quotation marks
c. a semi-colon
d. curly brackets

Points: 1 / 1

A 21. This modifier indicates that the method can be called from anywhere in a program, even from outside the class where the method is defined.

a. public
b. private
c. protected
d. static

Points: 1 / 1

B 22. The _____ modifier indicates that the method can be called only from inside the same class.

a. public
b. private
c. protected
d. static

Points: 1 / 1

A 23. Variables that are declared inside a subroutine are called _____.

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

Points: 1 / 1

C 24. It is considered to be good practice to make member variables and subroutines _____.

a. public
b. private
c. static
d. default

Points: 0 / 1

B 25. A static member variable that is declared to be final, is often referred to as a _____, since its value remains constant for the whole time the program is running.

a. default
b. named constant
c. enumerations
d. useless variables

Points: 1 / 1

C 26. Which of the following statements is true about the use of parameters with Java methods?

a. Methods without parameters can compile, but will not execute correctly.
b. All method declarations require parameters.
c. Many methods use parameters.
d. The use of parameters is optional to increase program readability.

Points: 1 / 1

C 27. What is true about a method declaration with multiple parameters?

a. All parameters must be the same data type.
b. All parameters must be different data types.
c. Parameter types may be the same or they may be different.
d. The parameter declarations depend on the method call.

Points: 1 / 1

B 28. What is the output of the following program?

```
public class Q24
{
    public static void main(String args [ ])
    {
      int n = 4;
      method1(n);
      method2(n + 2, 3);
      method3(n + n);
    }

    public static void method1(int x)
    {
      System.out.println("x = " + x);
    }

    public static void method2(int x, int y)
    {
      System.out.println("x + y = " + (x  + y) );
    }

    public static void method3(int n)
    {
      int x = 4;
      System.out.println("x = " + x);
    }

}
```

a.
```
4
63
8
```

b.
```
x = 4
x + y = 9
x = 4
```

c.
```
x = 4
x + y = 63
x = 8
```

d. Error message

Points: 1 / 1

C 29. What distinguishes the declaration of a void method?

a. The **public** keyword in the method heading
b. The **static** keyword in the method heading
c. The **void** keyword in the method heading
d. The **main** keyword in the method heading

Points: 1 / 1

D 30. 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 31. What distinguishes a call to a return method?

a. The method call is the only part of a complete program statement.
b. The method call provides a value, which is used in the program statement.
c. The method call includes the **void** keyword.
d. The method call includes the **return** keyword.

Points: 1 / 1

B 32. What is the output of the following program?

```
public class Q31
{
    public static void main(String args [ ])
    {
      int x = 25;
      int y = 10;
      System.out.println(x + " + " + y + " = " + Calc.add(x,y));
      System.out.println(x + " - " + y + " = " + Calc.sub(x,y));
      System.out.println(x + " * " + y + " = " + Calc.mul(x,y));
      System.out.println(x + " / " + y + " = " + Calc.div(x,y));
   }
}

class Calc
{
   public static int add(int p, int q)
   {
      int result = p + q;
      return result;
   }
   public static int sub(int p, int q)
   {
      int result = p - q;
      return result;
   }
   public static int mul(int p, int q)
   {
      return p * q;
   }
   public static int div(int p, int q)
   {
      return p / q;
   }
}
```

a.
```
25 + 10
25 - 10
25 * 10
25 / 10
```
b.
```
25 + 10 = 35
25 - 10 = 15
25 * 10 = 250
25 / 10 = 2
```
c.
```
35
15
250
2
```
d. Error message

Points: 1 / 1

 B 33. In the statement **`int num;`** ***int*** is the _____ and ***num*** is the _____.

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

data type
variable
method name
data type

Points: 1 / 1

 D 34. Which of the following programming features is part of Object Oriented Programming?

a. Encapsulation
b. Polymorphism
c. Inheritance
d. All of the above.

Points: 1 / 1

 B 35. What is the essence of encapsulation?

a. Combining data and the actions that access the data inside the same module.
b. Storing related data variables inside the same module.
c. Writing separate modules for separate tasks.
d. Error message

Points: 0 / 1

 C 36. The statement that *a square is a rectangle* is an example of

a. structured programming.
b. inheritance.
c. polymorphism.
d. encapsulation.

Points: 0 / 1

D 37. The actions in a Java class are called

a. methods.
b. procedures.
c. functions.
d. subroutines.

Points: 0 / 1

MULTIPLE RESPONSE

38. The parameters in the method call and the method heading must be the same;

a. quantity
b. sequence
c. type

Response: A, B, C
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 39. programming using various tools

Points: 1 / 1

A 40. a set of routines, protocols, and tools for building software applications

Points: 1 / 1

E 41. used to organize a group of classes

Points: 0 / 1

D 42. specific line of code that allows utilization of a class without using the full name

Points: 0 / 1

F 43. 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 44. single Java archive file that can contain many classes

Points: 1 / 1

A 45. system used to prepare most Java API documentation that can be used to create good API style documentation for any Java class

Points: 0 / 1

D 46. a special code that allows the programmer to use HTML commands

Points: 1 / 1

B 47. commands processed by the Javadoc tool

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 48. how a subroutine interacts with the rest of the program

Points: 1 / 1

F 49. a set conditions met before a subroutine is run

Points: 1 / 1

D 50. used to describe a member variable that is not visible, due to the scope of a local variable or parameter

Points: 1 / 1