Name: CAELAN KIMBALL **ID: 020022**
Date: 10/14/2019 **Class: Computer Programming 2**
Score: 38 / 44 (86.36%)

Chapter 5: Computer Programming 2

TRUE/FALSE

✓ F 1. True/False: The increment and decrement operators can be used in mathematical expressions; however, they cannot be used in relational expressions.

Points: 1 / 1

✓ F 2. True/False: A `while` loop is somewhat limited, because the counter can only be incremented by one each time through the loop.

Points: 1 / 1

✓ T 3. True/False: An initialization expression may be omitted from the `for` loop if no initialization is required.

Points: 1 / 1

✓ F 4. True/False: You may not use the `break` and `continue` statements within the same set of nested loops.

Points: 1 / 1

✓ F 5. True/False: The condition that is tested by a `while` loop must be enclosed in parentheses and terminated with a semicolon.

Points: 1 / 1

✓ F 6. True/False: You may not use the `break` statement in a nested loop.

Points: 1 / 1

✓ T 7. True/False: An output file is a file that data is written to.

Points: 1 / 1

✓ T 8. True/False: `string` objects have a member function named `c_str` that returns the contents of the object formatted as a null-terminated C-string.

Points: 1 / 1

MULTIPLE CHOICE

✓ B 9. These are operators that add and subtract one from their operands.

a. plus and minus
b. ++ and --
c. binary and unary
d. conditional and relational
e. None of these

Points: 1 / 1

✗ A 10. What is the output of the following code segment?

```
n = 1;

while (n <= 5)
    cout << n << '  ';
    n++;
```

a. 1 2 3 4 5
b. 1 1 1... and on forever
c. 2 3 4 5 6
d. 1 2 3 4
e. 2 3 4 5

Points: 0 / 1

✓ A 11. This operator increments the value of its operand, then uses the value in context.

a. prefix increment
b. postfix increment
c. prefix decrement
d. postfix decrement
e. None of these

Points: 1 / 1

✓ __A__ 12. The `while` loop has two important parts: an expression that is tested for a true or false value, and:

a. a statement or block that is repeated as long as the expression is true
b. a statement or block that is repeated only if the expression is false
c. one line of code that is repeated once, if the expression is true
d. a statement or block that is repeated once, if the expression is true

Points: 1 / 1

✓ __B__ 13. The `while` loop is this type of loop.

a. post-test
b. pre-test
c. infinite
d. limited
e. None of these

Points: 1 / 1

✓ __A__ 14. The statements in the body of a `while` loop may never be executed, whereas the statements in the body of a `do-while` loop will be executed:

a. at least once
b. at least twice
c. as many times as the user wishes
d. never
e. None of these

Points: 1 / 1

✓ __A__ 15. A `for` statement contains three expressions: initialization, test, and

a. update
b. reversal
c. null
d. validation
e. None of these

Points: 1 / 1

✓ __C__ 16. In a `for` statement, this expression is executed only once.

a. test
b. null
c. initialization
d. validation
e. None of these

Points: 1 / 1

✓ _C_ 17. You may define a ________ in the initialization expression of a `for` loop.

a. constant
b. function
c. variable
d. new data type
e. None of these

Points: 1 / 1

✓ _D_ 18. A loop that is inside another loop is called:

a. an infinite loop
b. a pre-test loop
c. a post-test loop
d. a nested loop
e. None of these

Points: 1 / 1

✗ _A_ 19. This statement may be used to stop a loop's current iteration and begin the next one.

a. break
b. terminate
c. re-iterate
d. continue
e. None of these

Points: 0 / 1

✓ _B_ 20. When the increment operator precedes its operand, as in `++num1`, the expression is in this mode.

a. postfix
b. prefix
c. preliminary
d. binary
e. None of these

Points: 1 / 1

✗ A 21. Look at the following statement.

```
while (x++ < 10)
```

Which operator is used first?

a. ++

b. <

c. Neither. The expression is invalid.

Points: 0 / 1

✓ B 22. The `while` loop contains an expression that is tested for a `true` or `false` value, and a statement or block that is repeated as long as the expression:

a. is `false`

b. is `true`

c. does not evaluate to `true` or `false`

d. evaluates to `true` or `false`

e. None of these

Points: 1 / 1

✓ D 23. This is a special value that marks the end of a list of values.

a. constant

b. variable

c. loop

d. sentinel

e. None of these

Points: 1 / 1

✗ A 24. What is the output of the following code segment?

```
n = 1;

for ( ; n <= 5; )
    cout << n << '  ';
    n++;
```

a. 1 2 3 4 5

b. 1 1 1 ... and on forever

c. 2 3 4 5 6

d. 1 2 3 4

e. 2 3 4 5

Points: 0 / 1

✓ B 25. The `do-while` loop is considered a(n) ________ loop.

a. pre-test
b. post-test
c. infinite
d. limited
e. None of these

Points: 1 / 1

✓ B 26. This is a pre-test loop that is ideal in situations where you do not want the loop to iterate if the condition is false from the beginning.

a. `do-while`
b. `while`
c. `for`
d. infinite
e. None of these

Points: 1 / 1

✓ B 27. This statement causes a loop to terminate early.

a. `stop`
b. `break`
c. `null`
d. `terminate`
e. None of these

Points: 1 / 1

✓ B 28. If you want a user to enter exactly 20 values, which loop would be the best to use?

a. `do-while`
b. `for`
c. `while`
d. `infinite`
e. None of these

Points: 1 / 1

✗ A 29. This statement may be used to stop a loop's current iteration and begin the next one.

a. break
b. terminate
c. return
d. continue
e. None of these

Points: 0 / 1

✓ B 30. What will the following loop display?

```
int x = 0;
while (x < 5)
{
    cout << x << endl;
    x++;
}
```

```
a. 0
   1
   2
   3
   4
   5
b. 0
   1
   2
   3
   4
c. 01 2 3 4
```

d. The loop will display numbers starting at 0, for infinity.

Points: 1 / 1

✓ A 31. What will the following code display?

```
int number = 6;

cout << number++ << endl;
```

a. 6 c. 7

b. 5 d. 0

Points: 1 / 1

✓ C 32. What will the following code display?

```
int number = 6;

cout << ++number << endl;
```

a. 6 c. 7

b. 5 d. 0

Points: 1 / 1

✓ A 33. What will the following code display?

```
int number = 6;
int x = 0;
x = number--;

cout << x << endl;
```

a. 6
b. 5
c. 7
d. 0

Points: 1 / 1

✓ B 34. What will the following code display?

```
int number = 6
int x = 0;
x = --number;

cout << x << endl;
```

a. 6
b. 5
c. 7
d. 0

Points: 1 / 1

✓ C 35. To allow file access in a program, you must `#include` this header file.

a. `file`
b. `fileaccess`
c. `fstream`
d. `cfile`

Points: 1 / 1

✓ D 36. This may be used to write information to a file.

a. `cout` object
b. `pen` object
c. `output` object
d. stream insertion operator
e. None of these

Points: 1 / 1

✓ D 37. To write data to a file, you define an object of this data type.

a. `outputFile`
b. `ifstream`
c. `fstream`
d. `ofstream`

Points: 1 / 1

✓ B 38. To read data from a file, you define an object of this data type.

a. `inputFile`
b. `ifstream`
c. `fstream`
d. `ofstream`

Points: 1 / 1

✓ C 39. Assuming `outFile` is a file stream object and `number` is a variable, which statement writes the contents of `number` to the file associated with `outFile`?

a. `write(outFile, number);` c. `outFile << number;`
b. `outFile >> number;` d. `number >> outFile;`

Points: 1 / 1

✓ C 40. Assuming `dataFile` is a file stream object, the statement:

```
dataFile.close();
```

a. is illegal in C++
b. needs a filename argument to execute correctly
c. closes a file
d. is legal but risks losing valuable data
e. None of these

Points: 1 / 1

✓ A 41. How many times will the following loop display "Hello"?

```
for (int i = 0; i < 20; i++)
    cout << "Hello!" << endl;
```

a. 20 c. 21
b. 19 d. An infinite number of times

Points: 1 / 1

✓ B 42. How many times will the following loop display "Hello"?

```
for (int i = 1; i < 20; i++)
    cout << "Hello!" << endl;
```

a. 20 c. 21
b. 19 d. An infinite number of times

Points: 1 / 1

✓ C 43. How many times will the following loop display "Hello"?

```
for (int i = 0; i <= 20; i++)
    cout << "Hello!" << endl;
```

a. 20 c. 21
b. 19 d. An infinite number of times

Points: 1 / 1

❌ B 44. How many times will the following loop display "Hello"?

```
    for (int i = 20; i > 0; i--)
        cout << "Hello!" << endl;
```

a. 20

b. 19

c. 21

d. An infinite number of times

Points: 0 / 1