Name: _____ **Class:** _____ **Date:** _____ **ID: A**

CST-186 Chapter 3 Study Guide

True/False
Indicate whether the statement is true or false.

____ 1. Numbers generated with the random module are not truly random.

____ 2. When an `if` structure's condition is `False`, its associated block is skipped.

____ 3. An `if` structure is useful for repeating a section of code.

____ 4. In Python, using indentation to indicate code blocks is encouraged but not required.

____ 5. Exactly one block of code of an `if-elif-else` structure will always execute

____ 6. The body of a while loop will execute at least once.

____ 7. Any value, of any type, can be treated as a condition.

____ 8. Liberal use of `break` and `continue` statements can help make the logic of a loop clearer.

____ 9. The value of the expression `not not not not False` is `False`.

____ 10. An algorithm is something you write after finishing a program as a way to document the code.

Multiple Choice
Identify the choice that best completes the statement or answers the question.

____ 11. In the following code, `random` is a(n) what?

```
print random.randrange(5)
```

a. module
b. function
c. argument
d. parameter

____ 12. In the following code, `randrange` is a(n) what?

```
print random.randrange(5)
```

a. module
b. function
c. argument
d. parameter

____ 13. What does the following code display?

```
import random
print random.randrange(5)
```

a. a random integer between 0 and 4, inclusive
b. a random integer between 0 and 5, inclusive
c. a random integer between 1 and 5, inclusive
d. five random integers

____ 14. Which line of code will generate a random integer from 5 to 9, inclusive? (Assume that random has been imported)

a. `random.randrange(5) + 5`
b. `random.randrange(4) + 5`
c. `random.randrange(5) + 4`
d. `random.randrange(9)`

____ 15. What will the following code display?

```
score = 100
if score = 100:
    print "Nice score",
print "End"
```

a. `Nice score`
b. `End`
c. `Nice score End`
d. `None of these`

____ 16. Which of the following is the not equal to comparison operator?

a. `=`
b. `==`
c. `!=`
d. `not`

____ 17. < is what kind of operator?

a. logical
b. comparison
c. arithmetic
d. file

____ 18. For what values of score will the string `"Good score"` be printed by the following code?

```
if score >= 100:
    if score <= 10:
        print "Good score"
```

a. 100 or greater
b. 10 or less
c. Between 10 and 100, inclusive
d. None of these

____ 19. For what values of score will the string "Good score" be printed by the following code?

```
if score <= 100:
    if score >= 10:
        print "Good score"
```

a. 100 or higher
b. 10 or lower
c. Between 10 and 100, inclusive
d. None of these

____ 20. What will the following code display?

```
score = 1200
if score > 1000:
    print "Wow",
if score > 500:
    print "Nice",
else:
    print "Sad",
```

a. Wow
b. Nice
c. Wow Nice
d. Wow Nice Sad

____ 21. What will the following code display?

```
score = 1200
if score > 1000:
    print "Wow",
elif score > 500:
    print "Nice",
else:
    print "Sad",
```

a. Wow
b. Nice
c. Wow Nice
d. Wow Nice Sad

____ 22. What does the following code display?

```
count = 0
while < 5
    print count,
```

a. 0 1 2 3 4
b. 1 2 3 4 5
c. 0 1 2 3 4 5
d. None of these

____ 23. How many times is `Yes` displayed by the following code?

```
count = 10
while count < 10:
    print "No"
    count += 1
print "Yes"
```

a. `0`
b. `1`
c. `9`
d. `10`

____ 24. What does the following code display?

```
count = 1
while count < 10:
    count += count
    if count == 8:
        continue
    print count,
```

a. `2 4 6`
b. `2 4 6 10`
c. `2 4 16`
d. `2 4 16 32`

____ 25. What does the following code display?

```
count = 2
while True:
    print count,
    count *= count
    if count > 100:
        break
```

a. `2 4 6`
b. `2 4 6 10`
c. `2 4 16`
d. `2 4 16 32`

____ 26. What will the following code display?

```
score = -1
if score:
    print score,
print score,
```

a. `-1`
b. `-1 -1`
c. `-2`
d. None of these

____ 27. `not` is what kind of operator?

a. logical
b. comparison
c. arithmetic
d. file

____ 28. How many times is the body of the following while loop executed?

```
x = 10
y = 1
while x > y and y < 10:
    x -= y
    y += 3
```

a. 0
b. 1
c. 2
d. 3

____ 29. How many times is the body of the following while loop executed?

```
x = 10
y = 1
while x > y or y < 10:
    x -= y
    y += 3
```

a. 0
b. 1
c. 2
d. 3

____ 30. Which of the following is true about step-wise refinement?
a. It makes algorithms longer
b. It turns algorithms into code
c. It makes algorithms shorter
d. It reverses algorithms

Completion
Complete each statement.

31. A(n) _____ operator compares two values. The result of such a comparison is either `True` or `False`.

32. A(n) _____ condition is a larger condition formed by joining simpler conditions.

33. A(n) _____ operator joins conditions to form a larger condition.

34. The _____ logical operator joins two conditions to form a single condition that's `False` only if both original conditions are `False`.

35. The _____ logical operator joins two conditions to form a single condition that's `True` only if both original conditions are `True`.

Matching

Match each item with a statement below

a. Dot notation
b. Branching
c. Condition
d. Block
e. Body
f. Tracing
g. Algorithm
h. Pseudocode
i. Stepwise refinement
j. Module

____ 36. A section of code associated with a loop that's executed while the loop's condition is `True`.

____ 37. A set of clear, easy-to-follow instructions for accomplishing some task.

____ 38. A convention used for accessing part of an object, such as a function from a module.

____ 39. A file that contains code meant to be used in other programs.

____ 40. The process used to rewrite algorithms in more detail so that they're ready for implementation.

____ 41. An expression that evaluates to either `True` or `False`.

____ 42. Examining the execution of a program and its internal values in single steps.

____ 43. A section of code indented to form a single unit.

____ 44. Something between English and a programming language.

____ 45. The act of a program taking one path of code instead of another.

Short Answer

46. How can random numbers make game programs more fun?

47. What are modules and how can using them help a programmer?

48. Why is it so easy to confuse the equal-to comparison operator and the assignment operator?

49. Why should you plan your programs?

50. What is stepwise refinement and how can it help a programmer?