LEARNING OUTCOMES AND ASSESSMENT MEASURES:

Students who successfully complete CSCI 192 will be able to:

1. Understand the fundamental concepts of a basic computer and its operations.
   TEST QUESTION #1:
   What are the 4 basic computer operations?
       a. data, information, input, output
       b. byte, process, output, storage
       c. load, store, execute, memory
       d. input, output, process, storage

   TEST QUESTION #2:
   What is the correct order of the instruction cycle?
       a. increment pointer, execute, interpret, fetch
       b. fetch, execute, interpret, increment pointer
       c. increment pointer, interpret, execute, fetch
       d. fetch, interpret, execute, increment pointer

   TEST QUESTION #3:
   What makes up a machine instruction?
       a. an opcode and output
       b. instructions
       c. operands
       d. an opcode and operands

   TEST QUESTION #4:
   What are the components of a computer?
       a. mouse, scanner, printer, memory
       b. input devices, output devices, communications devices, memory, processor
       c. motherboard, monitor, memory, input devices, output devices, processor
       d. storage devices, input devices, output devices, communications devices, memory, processor

2. Summarize the programming process.
   TEST QUESTION #5:
   What is the correct order of steps in the programming process?
       a. Understand, Plan, Code, Compile, Debug, Test, Put in production
       b. Plan, Code, Test, Debug, Put in production
       c. Understand, Plan, Code, Compile, Test, Debug, Put in production
       d. Put in production, Understand, Plan, Code, Compile, Test, Debug

3. Construct flowchart symbols and pseudocode statements.
   TEST QUESTION #6:
   What flowchart symbol is a parallelogram?
   a. input/output
   b. processing
   c. terminal
   d. decision

   TEST QUESTION #7:
   What flowchart symbol is a diamond?
   a. input/output
   b. processing
   c. terminal
   d. decision

   TEST QUESTION #8:
   What symbol represents a flowline?
   a. an arrow
   b. a rectangle
   c. a lozenge
   d. a circle

   TEST QUESTION #9:
   Which of the following statements displays the contents of a variable *number*, in pseudocode?
   a. get number
   b. print number
   c. cout << number << endl;
   d. load number

   TEST QUESTION #10:
   Which one of the following pseudocode statements correctly computes the average of five numbers?
   a. average = num1 + num2 + num3 + num4 + num5 / 5
   b. average = (num1 + num2 + num3 + num4 + num5 / 5)
   c. average = (num1 + num2 + num3 + num4 + num5) / 5
   d. average = num1 + num2 + num3 + num4 + num5 / (5)

4. Summarize the evolution of programming techniques.
   TEST QUESTION #11:
   Which of the following are programming techniques?
   a. object-oriented
   b. function-oriented
   c. procedural
   d. a and b
   e. a and c

   TEST QUESTION #12:
   Which one of the following is considered to be a low-level programming language?
   a. Java
   b. C++
   c. Visual Basic
   d. Assembler

TEST QUESTION #13:
Which one of the following is not a high-level programming language?
    a. C
    b. COBOL
    c. Windows XP
    d. Java

5. Comprehend object oriented programming concepts.
   TEST QUESTION #14:
   Object-oriented programming involves creating objects, describing their functions and attributes, and how they interact with other objects.
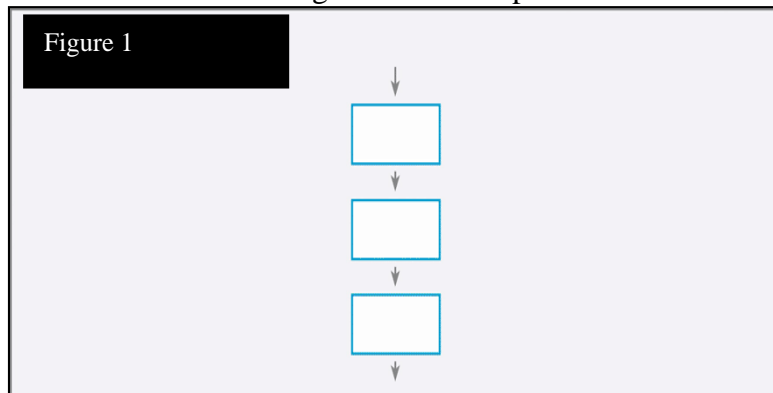   a. TRUE
    b. FALSE

6. Give examples of basic structures.
   TEST QUESTION #15:
   What are the three basic structures of any programming language?
    a. loop, stacking, nesting
    b. loop, selection, sequence
    c. selection, sequence, nesting
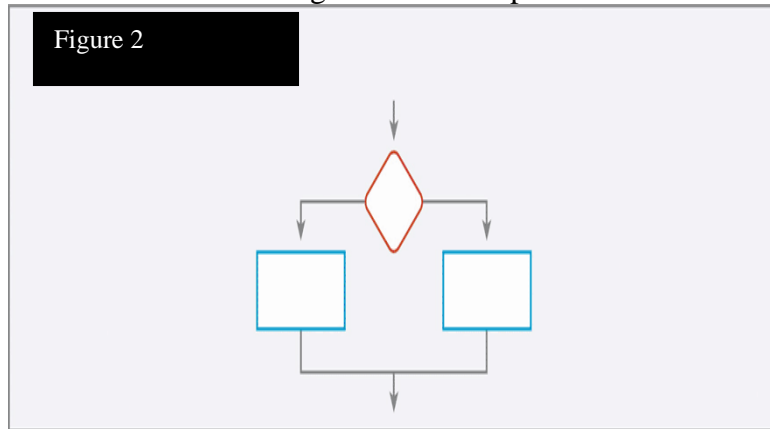   d. repetition, loop, sequence

TEST QUESTION #16:
Which structure does Figure 1 below represent?

Figure 1

    a. sequence
    b. selection
    c. loop

TEST QUESTION #17:

Which structure does Figure 2 below represent?

Figure 2

a. sequence
b. selection
c. loop

7. Design modules and hierarchy charts.
   TEST QUESTION #18:

   What does a hierarchy chart do?
   a. plans the logic of a program
   b. defines a location in memory
   c. illustrates relationships among modules
   d. gives instructions on maintaining a program

   TEST QUESTION #19:

   Why are modules useful?
   a. provide structure
   b. allow multiple programmers to work on a problem
   c. create friendly environment for the user
   d. both a and b
   e. both a and c

8. Document a program.
   TEST QUESTION #20:

   What is internal documentation?
   a. flowcharts
   b. comments
   c. pseudocode
   d. research papers

   TEST QUESTION #21:

   External documentation is supporting paperwork that programmers develop before they write their programs.
   a. TRUE
   b. FALSE

9. Explain logic in programming.
   TEST QUESTION #22:
   What is the output from the pseudocode below?

```
{
        float price = 98.69;
        float tax = 1.08;
        float total;

        total = price * tax;

        print total;
}
```
   a. 100.67
   b. 106.59
   c. 106
   d. 98.69

   TEST QUESTION #23:
   What is the output from the pseudocode below?

```
{
        float score1, score2, avg;
        string grade;

        score1 = 72.6;
        score2 = 93.0;
        avg = ( score1 + score2 ) / 2;
        if ( avg > 90 ) then
           grade = "A";
        else
           grade = "B";

        print grade;
}
```
   a. grade
   b. A
   c. B
   d. 82.8

TEST QUESTION #24:
What is the output from the pseudocode below?

```
{
        int x = 6, y = 1, z = 22;

        if ( x < 5 || x < -1) then
            print x;
        else-if ( y > 5 ) then
            print y;
        else
            print z;
}
```
    a. 6
    b. 1
    c. 22
    d. 5

10. Identify looping, control breaks and arrays.
TEST QUESTION #25:
What is the purpose of the return function?
    a. skips the body of a program
    b. terminates the execution of a loop
    c. returns zero to the operating system
    d. calls a function

TEST QUESTION #26:
What is an array?
    a. a datatype
    b. a loop structure
    c. a function that returns a value
    d. a variable that can store multiple values of the same type

---

BONUS: Write and design a complete program. (Email to csci10101@yahoo.com by 5:00 p.m., Friday)

Write a C++ program that displays the string literal "*Hello World!*" (without the quotes).

Task	Possible Points	Points Earned	Comments
File saved correctly	2		
Program compiles	15		
Comments included	3		
Output correct	5		
Code (readability)	5		
Total	**30**		