COSC175 - FINAL EXAM Review Topics

ipo, pseudocode, compile, debug, logic error, syntax error

Trace

Variables

K, MB, Byte, bit

Binary, octal, hex, decimal

Operators, +, -, \*, /, %, <, >, ==, AND, OR, NOT, precedence

Decisions

Loops –

While

counted

file, sentinel

Accumulator, counter

Functions, procedures,

arguments, formal vs actual

 pass by value, pass by reference

Scope - global, local

Hierarchy chart

Arrays

Linear search vs binary search

1. Describe in detail the steps to write a computer program. (Software Development Life Cycle)
2. Explain primary storage vs secondary storage. What happens when you boot the computer, what happens when you execute a program.
3. Describe structured programming.
4. Write an IPO for:      A program is required that will read in a tax rate (as a percentage) and the price of five items (use 5 variables). The program is to calculate the total price, before tax, of the items and then the tax payable on those items. The tax payable is computed by applying the tax rate percentage to the total price. Both values are to be printed as output.
5. Write a data declaration for each of the following:
	1. company name
	2. company address
	3. item price
	4. hourly pay rate
	5. number of employees
	6. 100 employee names
	7. 50 grades
	8. a 2 x 4 matrix (float)
6. What is a constant and when do you use them? Give an example.
7. Describe how integer and float values are stored. What is an overflow and how could it happen?
8. How is character data stored? How much space does it take up?
9. Why is the binary numbering system used?
10. Why is octal and hexadecimal used?
11. What is a bit vs a byte?
12. Convert 163 octal to binary.
13. Convert 16A hexadecimal to binary.
14. Convert 29 decimal to binary.
15. Convert 11100110 to octal
16. Convert 111100110 to hex
17. Convert 163 octal to binary.
18. Display 20 - 4 \* 2 - 8
19. Display  12 - 9 + 4 \* 2 + 3 ^ 2
20. Display 12 mod 7

 Assume the following hierarchy: NOT, AND, OR

Give the value of x, given A = FALSE, B = FALSE, C = FALSE, D = TRUE. E = 10, F = 15

1. x = A AND B
2. x = C OR D
3. x = NOT C
4. x = NOT (A AND D)
5. x = NOT C AND B
6. x = ( E < 10) AND (E = F)
7. x = A AND B OR C AND D

Write code for each of the following:

1. Prompt the user to enter a name.
2. Input a name from the user
3. Output Hello and the name you input.
4. Prompt for two numbers.
5. Input two numbers.
6. Compute the sum of the two numbers.
7. Output "the sum is" followed by the sum.
8. Modify the previous statement to echo print the two numbers entered. " the sum of *x* and *y* is *z*"; Do not use x, y, z use variables from above.
9. Input the price of an item.
10. Calculate the sales tax.
11. Compute the total price, given price and sales tax.
12. Compute the average of three test scores.
13. Increment a counter called numTests
14. Update an accumulator called sum with testScore
15. Input a number and print whether it is odd or even.
16. If the number is positive, add the number to posSum, else add the number to negSum.
17. Write a program to input three numbers and print out the smallest number.
18. Write the statements that allow you to input a series of numbers and print whether  it is positive or negative. Entering 0 causes the program to quit.
19. Write the statements to input numeric exam scores from a file and output the appropriate letter grade.
20. Write a program to print Towson University 10 times.
21. Write a program to print the following numbers:2 4 6 8 10 12(use a counted loop and a while loop)
22. Write a program that reads in integer grades from a file and computes the average.
23. Given an array: int stuff[100], initialize each element of stuff to -1.
24. Given an array: int stuff[100], input all elements from the user
25. Given an array: int stuff[100], sum the array
26. Given an array: int stuff[100], Print all elements of the array in reverse order
27. Given an array: int stuff[100], Find the smallest element in the array
28. Given an array: int stuff[100], search for the value 6
29. Given an array: int junk[10][20], display the array
30. Write a function called CalculateArea that calculates the area of a room when the width and length and area are passed as arguments. Are width and length and area passed by value or reference? Show a sample call.
31. Write a function called Area that calculates the area of a room and returns area when the width and length are passed as arguments. Show a sample call.
32. Write a function called CalculatePerimeter that calculates the perimeter when the width and length and perimeter are passed as arguments. Are width and length and perimeter passed by value or reference? Show a sample call.
33. Write a function called Perimeter that calculates the perimeter of a room and returns perimeter when the width and length are passed as arguments. Show a sample call.
34. Write a function called GetDimensions with two arguments width and length that prompts for and receives width and length.
35. Write a function called PrintDimensions with two arguments width and length that displays width and length.

**int stuff ;**

int **main ()**

**{**

 **int gamma;**

**gamma = 10 ;**

**stuff = 5 ;**

**cout <<  "gamma = " << gamma<<  "stuff = " << stuff << endl;**

**DoThis(gamma) ;**

**cout << "gamma = " << gamma << "stuff = " << stuff << endl;**

}

**void  DoThis( int alpha/\* in \*/)**

**{**

 **int stuff ;**

**cout << "alpha= " << alpha << "stuff= " << stuff << endl;**

**alpha = 3 ;**

**stuff = 4 ;**

**cout <<  "alpha = " << alpha << " stuff = " <<  stuff << endl;**

}

1. List any formal arguments.
2. List any actual arguments.
3. There are two variables called stuff. What is the scope of each?
4. What is the scope of gamma?
5. Give the output from executing the above program.

**int stuff ;**

int **main ()**

**{**

 **int gamma;**

**gamma = 10 ;**

**stuff = 5 ;**

**cout <<  "gamma = " << gamma<<  "stuff = " << stuff << endl;**

**DoThis(gamma) ;**

**cout << "gamma = " << gamma << "stuff = " << stuff << endl;**

}

**void  DoThis( int& alpha)**

**{**

 **int stuff ;**

**cout << "alpha= " << alpha << "stuff= " << stuff << endl;**

**alpha = 3 ;**

**stuff = 4 ;**

**cout <<  "alpha = " << alpha << " stuff = " <<  stuff << endl;**

}

80.  Give the output from executing the above program.