

COSC 1437 (DL)- Fall 2016

Quiz 2- Chapters 8-12 & Notes

Total Points: 60

Due: Sunday, October 23rd @ 11:59PM. Look at Syllabus/ICR about late work.

Directions: For Questions 1-26, clearly mark answers on a separate word (or notepad) document. See sample file/directions provided by your professor and submit to the appropriate location on the MyTCC (BlackBoard) site.

— Assume all variables are properly declared- unless otherwise mentioned.

Multiple Choice. Mark the one best answer for each question. (2 pts. each)

1. In order to use the string data type in a program, you must include the ____ directive in your program.
A. #include <iostream> C. #include <string>
B. #include <iomanip> D. None of the above
2. If the search item is the last element in the list, the algorithm makes ____ comparison(s).
A. one C. three
B. two D. n
3. A(n) _____ search is more efficient than a _____ search.
A. binary, linear C. integer, double
B. linear, binary D. None of these
4. Which line of the following program fragment contains a syntax error?

```
struct StatType          // Line 1
{                          // Line 2
    double height;        // Line 3
    int    weight;         // Line 4
}                          // Line 5
```

```
StatType stats;          // Line 6
```

- A. Line 1 C. Line 5
B. Line 3 D. Line 6

5. Consider the following program segment:

```
ifstream inFile;         //Line 1
int x, y;                 //Line 2

...                       //Line 3
inFile >> x >> y;         //Line 4
```

Which of the following statements at Line 3 can be used to open the file progdata.dat and input data from this file into x and y at Line 4?

- A. inFile.open("progdata.dat"); C. open.inFile("progdata.dat");
B. inFile.open, "progdata.dat"); D. open(inFile, "progdata.dat");

6. In addition to sequential access files, in C++ you can also create ____ files.
- A. random access
 - B. binary
 - C. Both A and B
 - D. None of the above
7. A struct variable can be passed as a parameter ____.
- A. only by const
 - B. only by reference
 - C. only by value
 - D. either by value or by reference
8. Consider the following statements:

```
struct supplierType
{
    string name;
    int supplierID;
};

struct paintType
{
    supplierType supplier;
    string color;
    string paintID;
};

paintType paint;
```

What is the data type of paint.supplier?

- A. string
 - B. paintType
 - C. supplierType
 - D. struct
9. The C++ operator ____ is used to destroy dynamic variables.
- A. destroy
 - B. delete
 - C. *
 - D. ~
10. Which of the following arithmetic operations is allowed on pointer variables?
- A. increment
 - B. modulus
 - C. multiplication
 - D. division
11. Executing:

```
string name = "Jamie Leonard";
name.replace(6, 7, "Kent");
```

changes the contents of the name variable to ____.

- A. "Kent"
- B. "Kent Jamie"
- C. "Jamie Kent"
- D. "JamieKent"

12. Executing:

```
string phone = "3120501111";  
phone.insert(0, "(");  
phone.insert(____, ")");    //Fill in the blank  
phone.insert(8, "-");
```

changes the contents of the phone variable to "(312)050-1111".

- | | |
|------|------|
| A. 3 | C. 5 |
| B. 4 | D. 6 |

13. What is the freebie you get automatically with an array?

- | | |
|---------------------------------------|---|
| A. Nothing is free in C++. | C. A *(reference) to the first element. |
| B. A *(pointer) to the first element. | D. A *prototype(). |

14. A selection sort is being used to arrange the following set of numbers in ascending order:

7 5 3 9 2 6

After the first pass of the sort is completed, what order will the data be in?

- | | |
|----------------|----------------|
| A. 2 5 3 9 7 6 | C. 2 3 5 6 7 9 |
| B. 5 3 7 2 6 9 | D. 5 7 3 9 2 6 |

Main Memory

	.
	.
	.
p 1200	1800
	.
	.
	.
num 1800	24
	.
	.
	.

```
int *p; int num;
```

p 1200

```
num 1800
```

- 4

Short Answer. Clearly mark answers as directed. Partial Credit will be given. (10 @ 2 each)

21. Write the statement to determine the length of the name string defined below and place the value into the number variable.

```
int number = 0;
string name = "Hello";
```

22. Given the declarations below, write the statements to combine the person's full name and store it into the full string variable.

```
string first = "First";
string middle = "Middle";
string last = "Last";
string space = " ";
string full = "";
```

23. Given the declarations:

```
enum WoodKind {HARDWOOD, SOFTWOOD};
struct Size
{
    int length;
    int width;
    int thickness;
};
struct Wood
{
    Size    dimensions;
    WoodKind kind;
    int     smoothSurfaces;
};
Wood oneBoard;
```

Write a statement to print the width of oneBoard.

24. Approximately how many comparisons are performed by a binary search of 1000 items if the search item is not in the list?

25. What is output by the following program fragment?

```
int  alpha = 35;
int  beta  = 50;

int* ptr1 = &alpha;
int* ptr2 = &beta;
ptr1 = ptr2;

cout << *ptr1 << ' ' << *ptr2 << endl;
cout << alpha << ' ' << beta << endl;
```

26. Write a C++ program that allows the user to enter the values for an array of integers, and select between sorting them in ascending or descending order. Print the unsorted and sorted array and the array after each pass. Use a function for the selection sort. Output should look similar to below.

Sample Run:

```
Enter the number of values in the array [integer]: 6
Enter element value 1: 11
Enter element value 2: 7
Enter element value 3: 23
Enter element value 4: 1
Enter element value 5: 16
Enter element value 6: 15

Enter sorting order [A for ascending, D for descending]: A

The unsorted values are
11 7 23 1 16 15

Pass 1: 1 7 23 11 16 15
Pass 2: 1 7 23 11 16 15
Pass 3: 1 7 11 23 16 15
Pass 4: 1 7 11 15 16 23
Pass 5: 1 7 11 15 16 23

The sorted values are
1 7 11 15 16 23
```

Fill in the missing parts of the program below to solve the problem as stated above. Do not add any additional lines of code. **(10 @ 2 each)**

```
//ShowSort.cpp

#include <iostream>
#include <iomanip>
#include <cctype>
using namespace std;

void selectionsort(int[], int, bool);
void showarray(const int[], int);

int main()
{
    int size, input, count = 0;
    char choice;
    bool choice2;

    const int defaultsize = 10;
    int value[defaultsize];

    cout << "Enter the number of values in the array [integer]: ";
    cin >> size;
```

```

do {
    cout << "Enter element value " << count + 1 << ": ";
    cin >> input;
    _____ //1
    count++;
}
while(count < size);

do {
    cout << "\nEnter sorting order [A for ascending, D for descending]: ";
    cin >> choice;
    cout << endl;

    switch(toupper(choice)) {
        case 'A':
            cout << "\nThe unsorted values are\n";
            _____ //2
            cout << endl;
            selectionsort(value,size,true);
            choice2 = false;
            break;
        case 'D':
            cout << "\nThe unsorted values are\n";
            showarray(value,size);
            cout << endl;
            _____ //3
            choice2 = false;
            break;
        default:
            cout << "\nPlease use correct letter\n";
            choice2 = true;
            break;
    }
}
while(choice2);

cout << "\nThe sorted values are\n";
showarray(value,size); //show the final result from sort
cout << endl;

return 0;
}

void selectionsort(int array[], int size, bool choice) {
    int startscan, minindex, minvalue;

    for(startscan = 0; startscan < (size - 1); startscan++) {
        _____ //4
        minvalue = array[startscan];

        for(int index = startscan + 1; index < size; index++) {
            if(choice == true){
                if (array[index] < minvalue) {

```

```

        minvalue = array[index];
        minindex = index;
    }
}
if (choice == false) {
    if (_____ ) { //5
        minvalue = array[index];
        minindex = index;
    }
}
}
array[minindex] = array[startscan];
array[startscan] = minvalue;

cout << "Pass " << startscan + 1 << ": ";
showarray(array,size);
}
}

void showarray(const int array[], int size) {
    for(int count = 0; count < size; count++)
        cout << array[count] << " ";
    cout << endl;
}

```

=====

Extra Credit: Implement the following program. Follows same program guidelines and graded on the same scale as program sets. Submit only your .cpp file- no test runs/folder required. Partial credit given. (10 points)

If two strings of equal length are superimposed on one another, then some letters may match. For example, consider the strings:

wonderwhowrotethebookonlove

AND

weallliveinayellowsubmarine

Note that there are three positions that contain the same letter: 1st, 14th, and 27th. Of the 27 possible positions, matches occur in three positions (11.1%). This percentage is called the index of coincidence for two strings. Write a C++ program that accepts two strings of the same length (error check) and determines their index of coincidence. Format the answer to one decimal place. Output should look similar to below.

Sample Run:

```

Enter the first string: Wonder who wrote the book on love
Enter the second string: weallliveinayellowsubmarine

```

```

The index of coincidence is: 11.1%

```

Name the program: IndexOfCoincidence.cpp, where XX are your initials.