

# **CSCI 131-- Techniques of Programming**

*College of the Holy Cross, Spring 2014*

## **Topics for Exam2:**

This sheet is intended to help you prepare for the second exam in this course. The exam will primarily focus on chapters 7 - 12 and 13.4 of the text book and up through lecture 25 and lab 8 (Except for the part on classes). The following topics have been covered since exam1. Each of the following topics may appear on the exam. In addition, any of the topics covered in the beginning of the course may also appear (see review sheet for exam1)

### 1. Functions

- Function declarations and definitions
- function calls
- void functions and functions that return a value
- parameters (pass by value, pass by reference)
- direction of data flow
- preconditions/postconditions for functions

### 2. Scope and Lifetime

- Scope rules
- Lifetime of variables
- local vs. global variables
- static variables

### 3. Simple Data Types

- integral and floating point types
- storage of data in memory
- operators (sizeof, cast, ternary, increment, decrement, combined assignment)
- the char type, ASCII code
- enumeration types

### 4. One Dimensional Arrays

- Declaring Arrays
- Accessing individual components of arrays
- Initializing arrays (in a declaration or with assignments)
- Passing arrays as parameters in functions
- Processing arrays
- Using parallel arrays
- Using indices with enum types

### 5. Strings

- Representation of strings in C++
- Initializing Strings
- Strings vs. characters
- String I/O
- String Library functions
- Using typedef with Strings and arrays of strings

## 6. Multidimensional arrays

- Representation as rows and columns
- Initializing a 2-dimensional array
- Processing rows (or Columns) of 2-D array
- Passing a 2-D array as a parameter
- Multidimensional (more than 2-D) arrays

## 7. Structs

- Declaring a struct type
- Accessing struct members
- Operations on struct types
- Passing structs as parameters
- Arrays of structs
- Hierarchical structs
- Unions

The following problems are intended to help you study for the exam.

1) Given the following enumeration:

```
enum Colors {RED, BLUE, GREEN, YELLOW, PURPLE};  
int colorRating[5] = {3, 1, 2, 5, 4};
```

a) What is the value of the expression: `colorRating[BLUE]` ?

b) Write a loop, using the enumeration `Colors` as indices, to print out the `colorRating` for each color.

2) A function, `SomeFunc`, has two formal parameters, `alpha` and `beta`, of type `int`. The data flow for `alpha` is one-way, into the function. The data flow for `beta` is two-way, into and out of the function. Write an appropriate heading for the function, `SomeFunc` (`SomeFunc()` does not return a value).

3) Consider the function definition:

```
void Demo(int& intVal, float floatVal) {  
    intVal = intVal *2;  
    floatVal = float(intVal) + 3.5;  
    return;  
}
```

Suppose the caller has variables myInt and myFloat whose values are 20 and 4.8 respectively. What are the values of myInt and myFloat after return from the following function call?

```
Demo(myInt, myFloat);
```

4) Given the function definition:

```
int Mystery( float someVal) {  
    if (someVal > 2.0) {  
        return 3 * int(someVal);  
    } else {  
        return 0;  
    }  
}
```

What is the value of the expression Mystery (4.2)?

5) a) Write a function,

```
float TenToThePower( int n)
```

that returns 10.0 raised to the integer power specified by n. Do **not** use the math.h library function, pow( ).

b) Write a C++ statement to compute 10.0 to the power of 3 and store the answer in the variable, tenCubed.

6) a) Write a nested for loop to produce the following output:

```
1
1 2
1 2 3
1 2 3 4
```

b) Write a function:

```
void Triangle (int n)
```

That produces a triangle like that above, with n rows. The first row prints out 1, the last row prints out the integers from 1 to n.

7) Given the following declarations:

```
struct ItemType {
    char description[40];
    int idNumber;
};
ItemType listOfProducts[10];
```

a) Write an expression that denotes the description of the third item in the list of products.

b) Write an expression that denotes the first character of the description of the tenth item in the list of products:

c) If the char type takes up 1 byte of memory, and the int type takes 2 bytes of memory, how many bytes of memory are allocated for the array, listOfProducts?

d) Write a C++ program fragment that will input (from standard input) the idNumbers for the entire list of products (don't forget to prompt the user).

8) Consider the following CString: "The rain in Spain".

a) Draw a diagram to show how this string would be stored in C++ memory.

b) What is the minimum length of the array needed to store this string?

9) Given the declaration

```
int checkers[8][8];
```

a) Write a statement that will store the number, 5, in the 2nd row and 3rd column of the array.

b) Write C++ code to store the numbers 1-8 in the first row, 9-16 in the second row, 17-24 in the third row, and so on.

10) Write a function to insert an integer element into a list of integers at a given index. The list of integers is stored in an array, but there are fewer integers in the list than the length of the array. The function should take 4 parameters: The array, the length of the list, the item to be inserted and the index where the item should be inserted. Assume that enough memory has been allocated so that the new length of the list will still fit into the allocated memory for the array (i.e. if the length parameter is 5, then at least 6 elements have been allocated for the array in memory).

11) Consider the following input data (from standard input):

One small step  
for a man

Given the following declarations:

```
char string1[20];  
char string2[20];  
char someChar;
```

a) Write code that will input the first line into string1 and the second line into string 2.

b) Write code that would input the first word into string1 and the second word into string2.

12) Consider the following declarations:

```
struct DateType {
    int month;
    int day;
    int year;
};
struct TimeType {
    int hours;
    int minutes;
    int seconds;
};
struct ShowTime {
    DateType showDate;
    TimeType curtainTime;
};
ShowTime myShowTime;
```

a) Write the C++ code to assign the appropriate values to myShowTime for a show that starts at 09:30:00 on the date 11/25/14.

b) Suppose there is a function, `GetDate(DateType& theDate)`, that will read in the date from standard input and store it in the parameter, `theDate`. Write the function call for `GetDate` that will read in the date for `myShowTime`.