

These are the types of questions that the exam may contain. This is not a sample exam. An actual exam may have harder or easier questions, and will cover other material as well.

1. Write a function that, given a float array parameter named `scores`, and an integer `size` that stores the length of the array, returns the average value in the array, or 0 if the array is empty.
2. Write a pseudocode description of a function that sorts an array. Use any sorting algorithm that you choose.
3. Write a function that, when given a string argument, returns a reversed copy of the string.
4. Write a function that will copy the contents of an input file stream to an output file stream. The function cannot assume that either stream is already open. The loop that actually does the copy should terminate on end of file. The function prototype should be

```
copy(istream& fin, ostream& fout);
```

5. Implement the following function description:

```
int countLetter ( char a[], int size, char letter) ;  
/*  
countLetter(a, s, ch) returns the number of occurrences of  
character ch among the first s characters in the string a,  
ignoring case. In other words, if ch is 'a', then 'A' and 'a'  
are each counted as matching.  
*/
```

6. Given the following class prototype, write the definitions of the two member functions and the constructor:

```
class cStudent  
{  
public:  
    cStudent ( char fn[], char ln[], int id = 0);  
    void mGetName (char name[], int MaxLength);  
    // After calling mGetName, name will contain the first name  
    // followed by the last name with a single space character  
    // in between, provided that MaxLength is large enough;  
    // otherwise it is empty.  
    void mChangeId (int i );  
    // replaces the old ID with the new one  
private:  
    char firstName[10];  
    char lastName[10];  
    int ID;  
}
```

7. Write a definition (not an implementation) of a class `Voxel` that encapsulates the representation of a point in the three dimensional space. A point is defined by its `x`, `y`, and `z` coordinates. The class must contain
8. Write the declarations and statements needed to read from a file named `datain.txt` in the same directory as the running program. Assume that the file contains one integer per line. Write the code that will add all of the numbers in the file and display the sum on the screen.

9. Write the declarations and statements needed to write to the file results.txt in the same directory as the running program. Assume that the array Grade[100] contains 100 float values. Write the code that will output the contents of this array into the file, one number per line.
10. You have implemented a class and the program that uses it in separate files. You just changed the class implementation file (but not the interface). What steps must be taken to bring the executable program up to date? In other words, what must be done, if anything, to each of the following files: class implementation file, class interface file, program file?
11. Write C++ code to create a dynamically allocated array of 100 floating point numbers named data.

12. Show the values of the array in the following program (1) before it enters the loop, (2) after it leaves the loop, and just before the program terminates.

```
#include <iostream>
using namespace std;
int main()
{
    int values[5] = {3};
    for (int i = 1; i < 5; i++)
    {
        values[i] = i;
    }
    values[0] = *(values + 1) + *(values + 4);
    return 0;
}
```

13. What is the output of the following program fragment?

```
char * b = new char [15];
char *c;
strncpy(b, "May 26, 2010");
c = strtok ( b, " ,");
while (c != NULL)
{
    cout << c << "\n";
    strtok( NULL, " ,");
}
```

14. For each statement below decide whether it is true (**T**) or false (**F**).

- (a) **T F** A friend function of a class `myClass` is a private member of `myClass` whose definition is not in `myClass`.
- (b) **T F** Functions can be passed to other functions as parameters using pointers.
- (c) **T F** A class can have more than one destructor in its definition: a default one and a user provided one.
- (d) **T F** The function whose prototype is `int f (int a, int b = 0, int c = 0, int d =0)`; can be called with three parameters.
- (e) **T F** Number of elements in an array can always be determined by using the `sizeof` operator on the name of the array and dividing it by the value returned by the `sizeof` operator applied to the name of the type of this array.