CS 161 Exam 2:

FORM 2 (Please put your name and form # on the scantron!!!!)

True (A)/False(B) (2 pts each):

1. The amount of memory used by an array depends upon the array's data type and how many elements in the array currently have data stored in them.
2. In C++, if you attempt to store more data in an array than it can hold, the compiler will issue an error.
3. You may use the exit() function to return the flow of control from a function back to main(), regardless of where the function was called from.
4. To account for the null terminator stored at the end of each C-string, the strlen function returns the number of characters in its argument, plus one.
5. Any algorithm that can be coded with recursion can also be coded using a loop.
6. A one-dimensional array can be initialized at the time it is defined, but a two-dimensional array cannot be.
7. An individual array element can be processed or passed to a function just like a regular C++ variable.
8. The following array definition is legal because C++ allows arrays to be implicitly sized.
   int grades[ ];
9. C++ allows arrays to have more than two dimensions.
10. The following statement is a valid C++ array definition.
    double money[25.00];
11. A pointer can be passed as an argument to a function.
12. The ampersand (&) is used to dereference a pointer variable in C++.
13. C-string can be assigned to an variable whose type is the string class.
14. C++ does not perform array bounds checking.

Multiple Choice (3 pts each)

15. The type of the literal string `"Hello"` is best described as
    a) `*char[]`.
    b) `*string`.
    c) `char *`.
    d) `string *`.
    e) None of the above

16. The statement `cout << &num1;` will output
    a) the value stored in the variable called `num1`.
    b) the string "`&num1`".
    c) the number 1.
    d) the memory address of the variable called `num1`.
    e) None of the above

17. To declare an array that will store students' last names of up to 25 characters in length, which is an appropriate statement?
    a) `char lastName[25];`
    b) `string lastName[25];`
    c) `string lastName[24];`
    d) `char lastName[26];`
    e) None of the above

18. An array can store a group of values, but the values must be
    a) constants.
    b) all the same data type.
    c) numeric, not characters or strings.
    d) declared at the time the array is created.
    e) none of the above.

19. What are the values in the array after execution of the following code?

```
int a[4] = {3, 7, 6, 2};
int i = 2;
a[i] = i + 1;
a[i + 1] = a[ i - 1];
a[1] = 5;
```

    a) 5, 3, 3, 6
    b) 3, 5, 3, 7
    c) 5, 7, 3, 7
    d) 5, 7, 2, 1

20. To use the `strlen` function in a program, you must `#include`
    a) `<iostring>`.
    b) `<stringlib>`.
    c) `<strlen>`.
    d) `<cstring>`.
    e) None of the above

21. Dynamic memory allocation occurs
    a) when a pointer fails to dereference the right variable.
    b) when a variable is created by the compiler.
    c) when a pointer is assigned an incorrect address.
    d) when a variable is created at run-time.
    e) None of the above

22. The statement int `*ptr = new int;` acquires memory to hold an integer and then
    a) assigns an integer value to the variable called `ptr`.
    b) initializes the allocated memory to `0`.
    c) creates a new pointer called `int`.
    d) sets `ptr` to point to the allocated memory.
    e) None of the above

23. The statement `cout << *ptr;` will output
    a) the address of the variable stored in `ptr`.
    b) the value stored in the address contained in `ptr`.
    c) the string `"*ptr"`.
    d) the address of the variable whose address is stored in `ptr`.
    e) None of the above

24. If dynamically allocated memory is not freed,
    a) a run-time error informs your user that the program did not free memory space.
    b) the source code will not link correctly.
    c) the system may run out of memory.
    d) it results in a compiler error.
    e) None of the above

25. An overloaded function is one
    a) that has too many parameters.
    b) that does different things depending on who calls it.
    c) that attempts to do too much in a single function.
    d) that call other functions.
    e) that has the same name as another function.

26. What is the value of pointer p after the following assignment?

```
p = new char;
```

a) 0
b) “”
c) stack address
d) heap address

27. When should a parameter be a reference parameter?

a) When the parameter is carrying information into the function that does not have to be returned
b) When the parameter is carrying information into the function that may be changed and the new value should be returned
c) When the information is to be returned from the function using the parameter.
d) Both b and c

28. What is the output of this code, given the following function definition?

```
int x =5, y = 2;
y = mixUp (x, y);
cout << x;

int mixUp  (int &p, int t)      //function definition
{
        p = p * t;
        return p + 1;
}
```

a) 5
b) 6
c) 10
d) 11

29. Given the function prototype and variable declarations, which of the following is a valid function call?

```
void compute (int, float, char&, int& );     // function prototype

int x, y;                    //variable declarations
float p, q;
char r, s;
```

a) compute (x, 7.3, ‘c’, y);
b) compute (y, p, s, x + y);
c) compute (5, p + q, r, y);
d) compute (x , s, r, 8);

30. A(n) _____ argument is one that is automatically passed to a parameter when the argument is left out of the function call.
    a) actual
    b) default
    c) floating-point
    d) null
    e) static

31. A recursive function should be designed to stop making recursive calls when it reaches its
    a) return statement.
    b) closing curly brace.
    c) last parameter.
    d) base case.
    e) None of the above

32. If the array defined as `int myArray[20][10]` is being passed to a function named `displayArray`, along with information on the number of rows and number of columns, which of the following function calls is correct?
    a) `displayArray(int myArray, 20, 10);`
    b) `displayArray(myArray, 20, 10);`
    c) `displayArray(myArray[20][10]);`
    d) `displayArray(myArray[ ][ ], 20, 10);`
    e) none of the above

33. The statement `int *ptr;` means
    a) the variable called `*ptr` will store an asterisk and an integer value.
    b) `ptr` is a pointer variable that will store the address of an integer.
    c) the variable called `ptr` will store an integer value.
    d) All of the above
    e) None of the above

34. Suppose that a recursive function with integer parameter `n` has a base case of 0, and for each non-base case, the function makes a recursive call with argument `n+1`. If the function is initially called with an actual argument of `n = 3`, the function call will
    a) return after a chain of 4 recursive calls.
    b) return after a chain of 2 recursive calls.
    c) return after a chain of 3 recursive calls.
    d) cause an infinite chain of recursive calls.
    e) None of the above

35. The correct reference for the element in the third row and fifth column of a matrix called myMatrix represented by a two dimensional array is:
    a) myMatrix [3] [5]
    b) myMatrix [2] [4]
    c) myMatrix [2, 4]
    d) myMatrix [5]

36. What is the output of the following segment of code?

```
int *p;
p = new int;
*p = 7;
cout << *p;
```

a) 0
b) 7
c) there will be an error message
d) we cannot tell because we do not know what memory address will be assigned to p

37. What is the value of b after the following function call?

```
int b = 3;
mystery (b);            // function call

void mystery  (int &val)      //function definition
{
        for (int c = 0; c < 5; c++)
                val += 2;
}
```

a) 2
b) 3
c) 13
d) 15

38. What is the output of the following function call, given the function definition below?

```
cout << tester (4);             // function call

int tester (int n)              // function definition
{
        if (n == 1)
                return 3;
        else
                return 2 * tester ( n – 1);
}
```

a) 3
b) 6
c) 12
d) 24

Extra Credit (2 pts each):

39. True(A)/False(B) Storage is allocated for a pointer and the data that it points to at the same time.

40. What would be the result of the call doTask (5, 4), given the following definition?

```
int doTask (int a, int b)
{
        if (a <= 2)
                return 5;
        else
                return doTask(a-1, b-1) + a + b;
}
```

a) 5
b) 10
c) 17
d) 26

41. Which of the following is a valid assignment, given the following declarations?

```
float *s;
float *t;
```

a) s = 50.0;
b) t = 2000;
c) s = s * 2;
d) s = t;

42. What is the output of the following code given the function definition below?

```
string word = "Hello";
mystery (word);
cout << word;

void mystery (string p)          // function definition
{
        int size = p.length ();
        for (int c = 0; c < size; c++)
                p.insert(0, "*");
}
```

a) Hello
b) *Hello
c) Hello*****
d) *****Hello

43. You are passing a two dimensional array, defined as below, to a function. What would be a correct function prototype? (ROWS and COLS are global constants.)

```
int table [ROWS] [COLS];
```

a) float calculate (int matrix [ ] [COLS], int rows);
b) float calculate (int matrix [ROWS] [ ], int rows);
c) float calculate (matrix [ROWS] [COLS], int rows);
d) float calculate (int matrix [ROWS] [ ], int cols);