

Exam 2 Review

1. Mark the following statements as true or false.

- a. The size of an array is determined at compile time.

True

- b. The declaration:

```
char str[] = "Sunny Day";
```

declares `str` to be a string of an unspecified length.

False

- c. As parameters, two-dimensional arrays are passed either by value or by reference.

True

- d. If a C++ function does not use parameters, parentheses around the empty parameter list are still required.

True

- e. When a `return` statement executes in a user-defined function, the function immediately exits.

True

- f. Because C-strings are stored in arrays, individual characters in the C-string can be accessed using the array component access notation.

True

- g. Parallel arrays are used to hold related information.

True

2. Consider the following function definition:

```
double func(double x, int y, string name)
{
}
```

Which of the following is the correct function prototype of the function `func`?

a. `double func();`

b. `double func(double, int, string);`

c. `double func(double x, int y, string name)`

d. `func(double x, int y, string name);`

3. Consider the following function prototypes:

```
int test(int, char, double, int);
```

```
double two(double, double);
```

```
char three(int, int, char, double);
```

Answer the following questions.

a. How many parameters does the function `test` have? What is the type of the function `test`?

Test has 4 parameters and is an integer function.

b. How many parameters does function `two` have? What is the type of function `two`?

Two has 2 parameters and is a double function.

c. How many parameters does function `three` have? What is the type of the function `three`?

Three has 4 parameters and is a character function.

d. How many actual parameters are needed to call the function `test`? What is the type of each actual parameter, and in what order should you use these parameters in a call to the function `test`?

4 parameters are needed to call test. Two ints, a char, and a double. They need to be called in the order (int, char, double, int).

e. Write a C++ statement that prints the value returned by the function `test` with the actual parameters `5`, `5`, `7.3`, and `'z'`.

`cout << test(5, 'z', 7.3, 5);`

f. Write a C++ statement that prints the value returned by function `two` with the actual parameters `17.5` and `18.3`, respectively.

`cout << two(17.5, 18.3);`

g. Write a C++ statement that prints the next character returned by function `three`. (Use your own actual parameters.)

`cout << three(1, 2, 'a', 3.14);`

4. Consider the following function:

```
int secret(int one)
{
    int i;
    int prod = 1;

    for(i = 1; i <= 3; i++)
        prod = prod * one;
    return prod;
}
```

a. What is the output of the following C++ statements?

i. `cout << secret(5) << endl;`

125

ii. `cout << 2*secret(6) << endl;`

432

b. What does the function `secret` do?

Raises the input to the 3rd power and returns it.

5. What is the output of the following program?

```
#include <iostream>
using namespace std;
int mystery(int x, int y, int z);

int main()
{
    cout << mystery(7, 8, 3) << endl;
    cout << mystery(10, 5, 30) << endl;
    cout << mystery(9, 12, 11) << endl;
    cout << mystery(5, 5, 8) << endl;
    cout << mystery(10, 10, 10) << endl;

    return 0;
}

int mystery(int x, int y, int z)
{
    if (x <= y && x <= z)
        return (y + z - x);
    else if (y <= z && y <= x)
        return (x + x - y);
    else
        return (x + y - z);
}
```

12

15

14

8

10

6. Write the definition of a void function that takes as input a decimal number and as output 3 times the value of the decimal number. Format your output to two decimal places.

```
void three(double num) {  
    cout << fixed << setprecision(2) << 3 * num << endl;  
}
```

7. Write the definition of a void function that takes as input two parameters of type int, say sum and testScore. The function updates the value of sum by adding the value of testScore. The new value of sum is reflected in the calling environment.

```
void update(int& sum, int testScore)  
{  
    sum += testScore;  
}
```

8. What is the output of the following program?

```
#include<iostream>

using namespacestd ;
void tryMe(int& v);
int main()
{
    int x = 8;
    for(int count = 1; count < 5; count++)
        tryMe(x);
    return 0;
}
void tryMe(int& v)
{
    static int num = 2;
    if(v % 2 == 0)
    {
        num++;
        v = v + 3;
    }
    else
    {
        num--;
        v = v + 5;
    }
    cout << v << ", " << num << endl;
}
```

11, 3

16, 2

19, 3

24, 2

9. Consider the following statement:

```
string str = "Now is the time for the party!";
```

What is the output of the following statements? (Assume that all parts are independent of each other.)

a. `cout << str.size() << endl;`

30

b. `cout << str.substr(7, 8) << endl;`

the time

c. `cout << str.insert(11, "best ") << endl;`

Now is the best time for the party!

d. `str.erase(16, 14);`
`str.insert(16, "to study for the exam?");`
`cout << str << endl;`

Now is the time to study for the exam?

10. What is the output of the following program?

```
#include<iostream>

using namespace std;

void find(int a, int& b, int& c);

int main()
{
    int one, two, three;
    one = 5;
    two = 10;
    three = 15;
    find(one, two, three);
    cout << one << ", " << two << ", " << three << endl;
    find(two, one, three);
    cout << one << ", " << two << ", " << three << endl;
    find(three, two, one);
    cout << one << ", " << two << ", " << three << endl;
    find(two, three, one);
    cout << one << ", " << two << ", " << three << endl;
    return 0;
}

void find(int a, int& b, int& c)
{
    int temp;
    c = a + b;
    temp = a;
    a = b;
    b = 2 * temp;
}
```

5, 10, 15
20, 10, 15
25, 30, 15
45, 30, 60

11. Identify error(s), if any, in the following array declarations.

a. `intlist[10];`

A space is needed: `int list[10];`

b. `const int size = 100;`
`double list[SIZE];`

Correct

c. `int numList[0..9];`

Not a valid size specification: `int numList[9];`

d. `string names[20];`

Correct

e. `scores[50] double;`

Incorrect placement of type: `double scores[50];`

12. What is the output of the following code?

```
int list[] = {6, 8, 2, 14, 13};
for(int i = 0; i < 4; i++)
    list[i] = list[i] - list[i + 1];
for(int i = 0; i < 5; i++)
    cout << i << " " << list[i] << endl;
```

```
0 -2
1 6
2 -12
3 1
4 13
```

13. What is stored in myList after the following C++ code executes?

```
double myList[5];
myList[0] = 3.0;
myList[1] = 4.0;
for (int i = 2; i < 5; i++)
{
    myList[i] = myList[i - 1] * myList[i - 2];
    if (i > 3)
        myList[i] = myList[i] / 4;
}
```

myList = [3 4 12 48 144];

14. Suppose that you have the following function definition.

```
void sum(int x, int y, int& z)
{
    z = x + y;
}
```

Consider the following declarations:

```
int list1[10], list2[10], list3[10];
int a, b, c;
```

Which of the following function calls is valid?

- a. **sum(a, b, c);**
- b. **sum(list1[0], list2[0], a);**
- c. **sum(list1[0], list2[0], list3[0]);**
- d. `sum(list1, list2, list3);`
- e. **sum(list1[0], b, 6);**
- f. `sum(list1, list2, c);`
- g. `for(int i = 1; i <= 10; i++)`
 `sum(list1[i], list2[i], list3[i]);`

15. Define a two-dimensional array named `temp` of three rows and four columns of type `int` such that the first row is initialized to 6, 8, 12, 9; the second row is initialized to 17, 5, 10, 6; and the third row is initialized to 14, 13, 16, 20.

```
int temp[3][4] = {  
    {6, 8, 12, 9},  
    {17, 5, 10, 6},  
    {14, 13, 16, 20},  
};
```

16. Suppose that you have the following declarations:

```
int times[30][7];
```

```
int speed[15][7];
int trees[100][7];
int students[50][7];
```

a. Write the definition of the function `print` that can be used to output the contents of these arrays.

```
void print(int list[][7], int rows)
{
    cout << "[\n";
    for (int i = 0; i < rows; ++i) {
        cout << "\t[";
        for (int j = 0; j < 6; ++j) {
            cout << list[i][j] << ",\t";
        }
        cout << list[i][6] << "], " << endl;
    }
    cout << "]\n";
}
```

b. Write the C++ statements that call the function `print` to output the contents of the arrays `times`, `speed`, `trees`, and `students`.

```
print(times, 30);
print(speed, 15);
print(trees, 100);
print(students, 50);
```

17. How would you specify how wide you want an output field of length 5?

`cout << setw(5);`

18. How would the following statements execute?

`val = 341.2576`

a. `cout << setprecision (3) << val;`

341

b. `cout << setprecision (3) << fixed << val;`

341.258

`val 2 = 32;`

c. `cout << setprecision (3) << showpoint << val2;`

32.0

19. Consider the following function prototype:

`void funcDefaultParam(double x = 7.3, int y = 4, string z = "");`

Which of the following function calls is correct?

a. `funcDefaultParam();`

b. `funcDefaultParam(2.8);`

c. `funcDefaultParam(3.2, 0, "h");`

d. `funcDefaultParam(9.2, "hi");`

e. `funcDefaultParam(7, 3);`

20. Given the declaration:

`char str1[15];`

`char str2[15]= "Good day";`

mark the following statements as valid or invalid. If a statement is invalid, explain why.

a. `str1 = str2;` **Invalid: cannot assign char arrays**

b. `if (strlen(str1) >= strlen(str2))`
`str1 = str2;` **Invalid: cannot assign char arrays**

c. `if(strcmp(str1,str2) < 0)`
`cout << "str1 is less than str2." << endl;` **Valid**

d. `if(str1 == str2)`
`cout << " Both strings are of the same length." << endl;`

Invalid: Comparing c-strings doesn't compare length

21. Which of the following functions should be defined as void?

- a. Return a sales commission, given the sales amount and the commission rate.
- b. Print the calendar for a month, given the month and year.**
- c. Return a square root for a number.
- d. Return a bool value indicating whether a number is even.
- e. Print a character a specified number of times.**

22. What is the output of the following code?

```
#include <iostream>
using namespace std;

int main()
{
    int matrix[4][4] = {{1, 2, 3, 4},
                        {4, 5, 6, 7},
                        {8, 9, 10, 11},
                        {12, 13, 14, 15}};

    for (int i = 0; i < 4; i++)
        cout << matrix[i][1] << " ";
    return 0;
}
```

- A. 1 2 3 4
- B. 4 5 6 7
- C. 1 3 8 12
- D. 2 5 9 13**
- E. 3 6 10 14

23. Write two versions of the function `mostFrequentCharacter`, one that accepts a C-string and one that accepts a string object, as its argument. The function should return the character that appears most frequently in the string. Demonstrate the function in a complete program. Do not use any string or cstring functions in the cstring version.

```
#include <iostream>
using namespace std;

char mostFrequentCharacter(char str[])
{
    int frequency[26] = {};
    int max_val = 0;
    int max_idx = 0;

    int i = 0;
    char cur_char;

    while ((cur_char = tolower(str[i])) != '\0') {
        if (cur_char >= 'a' && cur_char <= 'z') {
            ++frequency[cur_char - 'a'];
        }
        ++i;
    }

    for (int i = 0; i < 26; ++i) {
        if (frequency[i] >= max_val) {
            max_val = frequency[i];
            max_idx = i;
        }
    }

    return ('a' + max_idx);
}

char mostFrequentCharacter(string str)
{
    int frequency[26] = {};
    char cur_char;
    int max_val = 0;
    int max_idx = 0;

    for (int i = 0; i < str.size(); ++i) {
        cur_char = tolower(str[i]);
        if (cur_char >= 'a' && cur_char <= 'z') {
            ++frequency[cur_char - 'a'];
        }
    }

    for (int i = 0; i < 26; ++i) {
        if (frequency[i] >= max_val) {
            max_val = frequency[i];
            max_idx = i;
        }
    }
}
```

```
    return ('a' + max_idx);
}

int main()
{
    string str = "The quick brown fox jumps over the lazy dog";
    char cstr[] = "The quick brown fox jumps over the lazy dog";

    cout << "Freq String: " << mostFrequentCharacter(str) << endl;
    cout << "Freq C-str : " << mostFrequentCharacter(cstr) << endl;

    return 0;
}
```