Dan Nascimento

|  |  |
| --- | --- |
| Question | Points |
| 1 | 4 |
| 2 | 4 |
| 3 | 4 |
| 4 | 4 |
| 5 | 10 |
| 6 | 7 |
| 7 | 8 |
| 8 | 8 |
| 9 | 10 |
| 10 | 7 |
| 11 | 10 |
| 12 | 7 |
| 13 | 5 |
| 14 | 7 |
| 15 | 5 |

Intro to C++ - Test 1

1. The assignment operator in C++ ( x= 5, for instance) returns a value, **besides** copying that value to the variable. What is the value returned by (that is. what is the result of ) the assignment operator in this case?

The value returned by this assignment operator is 5.

1. Give an example of a cast from a double to an integer

    double x = 1.5;

    int y = static\_cast<int>(x);

1. What is the difference in C++, if any, between the expressions:

3 + 3/2 and 3 + 3.0/2 ? IF THE RESULT IS DIFFERENT, THEN EXPLAIN WHY

The difference is that 3 + 3/2 results to 4 and 3 + 3.0/2 results to 4.5  
This is because the first expression uses arithmetic operators with integers which will truncate the result of 3/2 to 1, then 3 + 1 is 4. The second expression uses arithmetic operators with a floating-point value, which leaves the decimal point so 3 + 1.5 is 4.5

1. Write in C++ the Boolean operation: “a less than c or a not equal to b”

    (a < c) || (a != b)

1. Write a function *max* that returns the greater of two integers. For instance

*cout << max (3, 5);*

should display 5. (That is, the function should RETURN the greatest of the two values)

int max(int a, int b)

{

    if (a > b) {

        return a;

    } else {

        return b;

    }

}

1. Rewrite the following function so that the new version produces the same output given

the same input as the one presented below. The new version should have ONLY ONE return statement.

*int example (int a)*

*{*

*if (a > 0)*

*return a;*

*if (a < 0 )*

*return -1\*a;*

*if (a == 0)*

*return 0;*

}

int example (int a)

{

    int output;

    if (a > 0)

        output = a;

    if (a < 0)

        output = -1\*a;

    if (a == 0)

        output = 0;

    return output;

}

1. Write an *if..else* statement equivalent to:

" If c is less than or equal to d then check if c is greater than 5 and if so then print c, but if c is neither less than nor equal to d, print d."

**ONLY two if’s are allowed**

int ifstatement(int c, int d)

{

    if (c <= d)

    {

        if (c > 5) {

            cout << c << endl;

        }

    } else {

        cout << d << endl;

    }

}

1. A student is asked to write a switch whose behavior is as follows:

The switch is controlled by a variable of type char. There are two cases: if the char is a vowel ('a', 'e', 'i', 'o', 'u'), the entered vowel will be displayed; otherwise, the message "Not a vowel" will be displayed.

When the assignment is turned in, the instructor notices that upon entering the letter 'e', the letter 'i' is also displayed. This does not happen when entering any other vowel, that is, if any of 'a', 'i', 'o', 'u' is entered, only the input vowel will be displayed.

Complete the *switch* statement below so that the behavior described above is preserved. That is, when entering ‘e’, the letter ‘i’ is also displayed. Make sure that each case in the switch only displays one letter.

The structure of the program is:

*char c;*

*cin >> c; //enter a char from the keyboard*

*switch (c)*

*{*

# //YOUR CODE HERE

*}*

I preserved the mistake the student made by not putting a break after case 'e':

    char c;

    cin >> c;

    switch (c)

    {

        case 'a':

            cout << 'a' << endl;

            break;

        case 'e':

            cout << 'e' << endl;

        case 'i':

            cout << 'i' << endl;

            break;

        case 'o':

            cout << 'o' << endl;

            break;

        case 'u':

            cout << 'u' << endl;

            break;

        default:

            cout << "Not a vowel\n";

            break;

    }

1. Complete the program fragment below so that the loop displays a sequence of integers between 7 and 0:

*int n* = YOUR ENTRY HERE;

*while (n >= 0) {*

*cout << n << ‘ ‘;*

*//****YOUR ENTRY HERE***

n--;

*}*

With n--; the loop will print n and decrement n by 1 each loop until n is no longer greater than or equal to 0.

1. Given the following incomplete program, write the missing code so that a two-digit integer *number* entered from the keyboard is split into its high and low digits using the division and remainder operations.

*#include< iostream>*

*using namespace std;*

*int main()*

*{*

*int number, high, low;*

*cout << " Enter a two-digit integer" << endl;*

*cin >> number;*

# //YOUR CODE HERE

*cout << high << ' ' << low << end;*

*return 0*;

*}*

#include <iostream>

using namespace std;

int main()

{

    int number, high, low;

    cout << " Enter a two-digit integer" << endl;

    cin >> number;

    //YOUR CODE HERE

    if ((number / 10) > (number % 10)) {

        high = number / 10;

        low = number % 10;

    } else {

        high = number % 10;

        low = number / 10;

    }

    cout << "high: " << high << ' ' << "low: " << low << endl;

    return 0;

}

11. Write a function that returns the greatest number out of a sequence of three integers. Therefore, a call like (as an example):

*greatest (4, 7, 5)* should return 7. Make sure your implementation returns the correct value if the three numbers entered happen to be the same number. Remember: returning a value is NOT the same as displaying a value.

int greatest(int a, int b, int c)

{

    int num;

    if (a >= b && a >= c) {

        num = a;

    } else if (b >= a && b >= c) {

        num = b;

    } else {

        num = c;

    }

    return num;

}

12. Change this function so that it uses only one *if..else* to accomplish the same:

*string example (string a, string b)*

*{*

*if (a < b)*

*return a;*

*if (a > b)*

*return b;*

*if (a == b)*

*return a;*

*}*

string example(string a, string b)

{

    if (a < b)

        return a;

    else if (a > b)

        return b;

    else

        return a;

}

13. Convert to decimal de binary number 10110110. Show the process

(1 \* 2^7) + (0 \* 2^6) + (1 \* 2^5) + (1 \* 2^4) + (0 \* 2^3) + (1 \* 2^2) + (1 \* 2^1) + (0 \* 2^0)  
128 + 0 + 32 + 16 + 0 + 4 + 2 + 0 = 182

14. Write the formatting code, using setw, setfill, right or left, so that the following output is displayed:

Within 15 columns, display your first name left justified and complete the remaining columns with the fill character ‘\*’ . The fill character \* must appear only once in the code.

For example, if the first name is Clarence:

Clarence\*\*\*\*\*\*\*

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

int main()

{

    string firstName = "Dan";

    cout << setw(15) << left << setfill('\*') << firstName << endl;

    return 0;

}

15. Define a input file stream variable to associate with an actual file named *myfile.txt*

so we can read from that file.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

    ifstream inFile;

    inFile.open("myfile.txt");

    return 0;

}