

Name: Otto Wiking Høyer

Spring 2024

Date: 2/5-24

CSCI 1300: Recitation 3

Please make sure to write your name and the date in the top left corner. You may use any course materials to answer the following questions and you may collaborate with others, but the work shown must be your own.

1 Spot The Error

Problem 1.1. The program below displays a message about the weather. Identify the error(s) and write the correct line(s).

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

int main()
{
    int humidity = 44;
    int temperature = 25;

    if(humidity > 55)
    {
        if(temperature >= 35)
        {
            cout << "The weather is hot and humid." << endl;
        }
        else
        {
            cout << "The weather is cold and humid." << endl;
        }
    }
    else
    {
        if(temperature > 35)
        {
            cout << "The weather is hot and dry." << endl;
        }
        else
        {
            cout << "The weather is cold and dry." << endl;
        }
    }

    return 0;
}
```

Name:

CSCI 1300: Recitation 3

Spring 20

Date:

Problem 1.2. The program below contains a function, `makeGreeting`, which is supposed to take in two words and merge them. It returns this new string to the main function, which then prints it. Identify the error(s) and write the correct line(s).

```
#include <iostream>
#include <string>
using namespace std;
double string makeGreeting(string firstword, string secondword)
{
    string merged_greeting = firstword + " " + secondword;
    return merged_greeting;
}

int main()
{
    string new_greeting = makeGreeting("Good", "Day");
    cout << new_greeting << endl;
}
```

Problem 1.3. The program below will display the average between two values by calling the function `avg`. The correct output should be 10.5. Identify the error(s) and write the correct line(s).

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

int main()
{
    double average = avg(15, 6);
    cout << average << endl;
    return 0;
}

double double double double
int avg(int a, int b)
{
    int x = (a+b) / 2.0;
    return x;
}
```

Name:

Spring 2024

Date:

CSCI 1300: Recitation 3

Problem 1.4. The program below will calculate and display the length of the hypotenuse of a right triangle given the length of two sides. This is done by calling the function `calculateHypotenuse`. Identify the error(s) and write the correct line(s).

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

int double calculateHypotenuse(int side1, int side2)
{
    cout << "Enter side 1: " << endl;
    cin >> side1;
    cout << "Enter side 2: " << endl;
    cin >> side2;

    double hypotenuse = sqrt(pow(side1, 2) + pow(side2, 2));

    return hypotenuse;
}

int main()
{
    double hypotenuse = calculateHypotenuse(3, 4);
    cout << hypotenuse << endl;
    return 0;
}
```

Problem 1.5. The program below aims to calculate and display the decimal equivalent of the reciprocal for a user-input value between 1 and 10. Identify the error(s) and write the correct line(s)

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

double findReciprocal(int x)
{
    double ans;
    ans = (1.0/num);
    return ans;
}

int main()
{
    int num;
    double ans;
    cout << "Enter a value between 1 and 10" << endl;
    cin >> num;
    ans = findReciprocal(num);
    cout << "The associated fraction can be written as " << ans << endl;
    return 0;
}
```

Name:

Spring 2024

Date:

CSCI 1300: Recitation 3

Problem 1.6. The program below contains correctly working code that determines the letter grade corresponding to a given score. There are no syntax or logic errors in this code. However, it has multiple style errors making the code very difficult to read. These errors can range from usage of unintended white space to having extraneous variables or clauses in your code. Rewrite the below code to improve readability.

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

int main()
{
    double score = 0;
    char grade='F';
    cout << "Enter the score you earned for CSCI 1300: ";
    cin >> score;

    if (score < 60)
    {
        cout << "Grade: F" << endl;
    }
    else if (score <= 62.99) { Move down curly bracket to next line
        cout << "Grade: D-" << endl;
    }
    else if (score <= 66.99)
    {
        cout << "Grade: D" << endl;
    }
    else if (score >= 67 && score <= 69.99) rewrite conditions
    {
        cout << "Grade: D+" << endl;
    }
    else if (score >= 70 && score <= 72.99)
    {
        cout << "Grade: C-" << endl;
    }
    else if (score >= 73 && score <= 76.99)
    {
        cout << "Grade: C" << endl;
    }
    else if (score >= 77 && score <= 79.99) add spaces
    {
        cout << "Grade: C+" << endl;
    }
    else if (score >= 80 && score <= 82.99)
    {
        cout << "Grade: B-" << endl;
    }
    else if (score >= 83 && score <= 86.99)
    {
        cout << "Grade: B" << endl;
    }
    if (score >= 87 && score <= 89.99) {
        cout << "Grade: B+" << endl;
    }
    else if (score >= 90 && score <= 92.99) { enter
        cout << "Grade: A-" << endl;
    }
    else if (score >= 93)
    {
        cout << "Grade: A" << endl;
    }
    return 0;
}
```

move curly bracket

Name:

Date:

CSCI 1300: Recitation 3

Spring 2024

2 Coffee Shop

You are a coffee shop owner, managing multiple coffee machines. The water for brewing coffee is sourced from the water compartment of the coffee machine which needs to be manually refilled. Design two functions to monitor the amount of water you use and refill. Both functions should take in your current water level in the compartment and an amount to be used from or refilled to the compartment, and then return the new water level in your compartment. Here are two function headers to get you started:

```
double useWater(double current_level, double amount)
double refillWater(double current_level, double amount)
```

There are some important details to consider: you cannot use more water than what you have in your central tank, and you cannot use or restock with a negative amount. As such, here are the details for both of these functions:

Function: useWater(double, double)	<code>double useWater(double current_level, double amount)</code>
Purpose:	Monitor the water level in the compartment when you use a certain amount of water out of it
Parameters:	<code>current_level</code> - a double that represents current level in the central tank <code>amount</code> - a double that represents the amount of water you want to use
Return value:	It returns the new water level in the compartment
Error handling/ Boundary conditions:	<ul style="list-style-type: none">• If the argument for <code>current_level</code> < 0, 0 is returned• If the argument for <code>amount</code> < 0, <code>current_level</code> is returned• If $(\text{current_level} - \text{amount}) < 0$, <code>current_level</code> is returned
Example:	<div>Sample Code 1:<pre>int main() { double current_level = 71.4; double amount = 20; double result = useWater(current_level, ↪ amount); cout << "The new water level is " << ↪ result << endl; return 0; }</pre></div> <div>Sample Output 1: The new water level is 51.40</div>

Name:

Spring 2024

Date:

CSCI 1300: Recitation 3

Function: refillWater(double, double)	<code>double refillWater(double current_level, double amount)</code>
Purpose:	Monitor the water level in the compartment when you refill a certain amount of water to it
Parameters:	<p><code>current_level</code> - a double that represents current level in the central tank</p> <p><code>amount</code> - a double that represents the amount of water you want to refill</p>
Return value:	It returns the new water level in the compartment.
Error handling/ Boundary conditions:	<ul style="list-style-type: none"> • If the argument for <code>current_level</code> < 0, 0 is returned • If the argument for <code>amount</code> < 0, <code>current_level</code> is returned
Example:	<div> <p>Sample Code 2:</p> <pre>int main() { double current_level = 100; double amount = 20.5; double result = refillWater(current_level, amount); cout << "The new water level is " << result << endl; return 0; }</pre> </div> <div> <p>Sample Output 2:</p> <p>The new water level is 120.50</p> </div>

Problem 2.1. Write out the steps you would use to solve this problem by hand as pseudocode.

use Water

declare variables
get input values.
check if value is reasonable
if it is, calculate resulting value
return waterlevel

Refill Water

declare variables
get input values for variables
check if inputs are reasonable
if it is, calculate resulting waterlevel
return waterlevel

Name:

Date:

CSCI 1300: Recitation 3

Spring 2024

Problem 2.2. Pick a possible value for `current_level` and `amount`. Follow the steps you wrote for these numbers for `useWater` to find your end result, and verify it.

`current_level = 3`
`amount = 0.5`

returns 2.5

Problem 2.3. Pick different possible values for `current_level` and `amount`. Follow the steps you wrote for these numbers for `refillWater` to find your end result, and verify it.

<code>current_level</code>	<code>amount</code>	result
3	1	2
3	2	1
1	0.5	0.5

Problem 2.4. Identify two possible values that are 'boundaries' in this problem that you will have to test. What should happen for these values? Is it different for `useWater` and `refillWater`?

If `current_level = amount`

should be ok

If `current_level = 0`

should be ok for refill, not for use

Problem 2.5. Translate your pseudocode into a c++ program to solve the above code.