

Name:

Spring 2024

Date:

CSCI 1300: Recitation 4

```
int main()
{
    assert(idLengthCheck("12345678"));
    assert(idLengthCheck("123456789") == false);
    return 0;
}
```

erase

Problem 1.4. The program below will use two functions: one to check for password match and another to check if the ID is valid before registering the user. Assume the relevant functions have been defined successfully. Identify the error(s) in the code below, and write the correct line(s).

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

```
bool passwordMatchCheck(char, char);
bool idLengthCheck(char);
```

Strings for these

```
int main() {
    int ID;
    string password;
    string confirmPassword;

    cout << "Enter your member ID: ";
    cin >> ID;
    assert(idLengthCheck(ID));

    cout << "Enter your password: ";
    cin >> password;

    cout << "Confirm your password: ";
    cin >> confirmPassword;

    if (passwordMatchCheck(password, confirmPassword))
    {
        cout << "Password set successfully for " << username << "." << endl;
    }
    else if (!passwordMatchCheck(password, confirmPassword))
    {
        cout << "Passwords do not match." << endl;
    }
    else if (!idLengthCheck(ID))
    {
        cout << "ID is invalid." << endl;
    }

    return 0;
}

bool passwordMatchCheck(string password, string confirmPassword)
{
    // appropriate definitions
}

bool idLengthCheck(string password)
{
    // appropriate definitions
}
```

Name:

Spring 2024

Date:

CSCI 1300: Recitation 4

Problem 1.5. The program below is a working program that uses the `getPrice` function to compute the price of a wall frame of a given area and color. This code does not contain any syntax or logical errors. 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. Identify the style error(s) in the code below and rewrite the below code to improve readability.

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

double getPrice(double area, string color) {
    assert(area>0); double cost = 0.0;
    if (color == "green") {
        cost = 4; }
    ← Enter
    else if (color == "red")
    { cost = 3; } 2
    ← else if (color == "orange")
    {
        cost = 2;
    }
    ← else if (color == "blue")
    {
        cost = 1;
    }
    } return area * cost; }

int main()
{
    string color, shape;
    int area_choice;
    double radius;
    double area = 0;

    cout << "Enter the area of the frame: (1) 5x5 (2) 4x6 (3) 8x10" << endl;
    cin >> area_choice;
    assert(
        ← area_choice == 1 || area_choice == 2 || area_choice == 3
    );
    ← move back
    if (area_choice == 1) { area = 5*5; } ← own line
    else if (area_choice == 2) { area = 4*6; } ←
    else if (area_choice == 3) { area = 8*10; } ←

    cout << "Enter the color of the frame: (green, red, orange, blue): ";
    cin >> color;
    ← one line
    { assert(
        color == "green" || color == "red" || color == "orange" || color == "blue"
    ); }

    double price = getPrice(area, color);

    cout << "You will receive a "<< color << " color frame with a price of $" << price
    ← << ". "; End!
    cout << "Thank you for your business." << endl;

    return 0;
}
```

Name:

CSCI 1300: Recitation 4

Date:

2 Coasters

The program below is created by a coaster-selling company with a thoughtful gesture to provide customers with personalized coasters on their birthdays. To achieve this, the company requests the customer's birth month and their preferred coaster shape(circle or square). The program then determines the coaster's color and its price, which is then displayed to the customer. Each color has a value associated with it. The area of the coaster and this value is used to identify the price of the coaster. The snippet of code below is provided in the file rec4.cpp to use in VS Code!

```
#include <iostream>
#include <string>

using namespace std;

/**
 * @brief Function to determine the color associated with their birth month
 * @param birthMonth of the user
 * @return string - color associated with their birth month
 */
string getColor(int birthMonth)
{
    // Check the birth month and return the corresponding color
    if (birthMonth >= 1 && birthMonth <= 3)
    {
        return "green";
    }
    else if (birthMonth >= 4 && birthMonth <= 6)
    {
        return "red";
    }
    else if (birthMonth >= 7 && birthMonth <= 9)
    {
        return "orange";
    }
    else if (birthMonth >= 10 && birthMonth <= 12)
    {
        return "blue";
    }
    else
    {
        return "";
    }
}

/**
 * @brief Function to calculate the area of the coaster based on shape(square or
 * circle)
 * @param shape of the coaster that user specified
 * @param size of the coaster, which is either the side length (for a square) or the
 * radius (for a circle)
 * @return double - area of the coaster based on the shape
 */
double getArea(string shape, double size)
{
    const double pi = 3.14159;
    if (shape == "square")
    {
        return size * size;
    }
}
```


Name:

Spring 2024

Date:

CSCI 1300: Recitation 4

```
    else if (shape == "circle")
    {
        return pi * size * size;
    }
    else
    {
        return 0.0;
    }
}

/**
 * @brief Function to calculate the price of the coaster based on area and color, where
 * → each color has a specific cost associated with it
 * @param area of the coaster
 * @param color of the coaster
 * @return double - price of the coaster based on area and color
 */
double getPrice(double area, string color)
{
    double cost = 0.0;
    if (color == "green")
    {
        cost = 4;
    }
    else if (color == "red")
    {
        cost = 3;
    }
    else if (color == "orange")
    {
        cost = 2;
    }
    else if (color == "blue")
    {
        cost = 1;
    }
    return area * cost;
}
```

You now need to work through these functions and understand what each one does. You will then test these functions to make sure they work as expected.

Name:

Spring 2024

Date:

CSCI 1300: Recitation 4

Problem 2.1. You have seen function tables designed to convey the information in a function on previous assignments. Now, given the code above, fill in the following table for the `getColor` function. This table does not have an "Example" row because you will work through that in the subsequent problems.

Function: <code>getColor(int)</code>	<code>String getColor (int birthmonth)</code>
Purpose:	Give a color from birthmonth
Parameters:	birthmonth - an int
Return value:	String a string corresponding to a color
Error handling/ Boundary conditions:	If input is an int, If input is $\in [1, 12]$

Problem 2.2. Identify the boundary conditions for `getArea` and `getPrice` below.

Area is "square" or "circle" otherwise error

Size is a positive number, otherwise error

area is > 0

String is a color

Name:

Date:

CSCI 1300: Recitation 4

Spring 2024

Problem 2.3. Pick four sample inputs for each of the three functions above (twelve total). Make sure that you cover the boundary conditions with these sample runs, as well as a few values in the expected range. Determine what the expected values should be for each one.

$1 \rightarrow \text{"green"}$ $0 \rightarrow \text{error}$ $-1 \rightarrow \text{error}$	$\text{"triangle"} \rightarrow \text{error}$ $\text{size} = 0 \rightarrow \text{error}$ $\text{size} = 1 \rightarrow 1 \text{ or } \pi$
$\text{area} = 0 \rightarrow \text{error}$ $1 \text{ red} \rightarrow 3$ $2 \text{ blue} \rightarrow 2$	

Problem 2.4. Translate each of the sample inputs and expected outputs you created into assert statements.

```
assert(get_color(1) == "green")  
assert(birthmonth >= 1)  
assert(birthmonth != 0)
```

Problem 2.5. Copy the functions above into VS Code and write a main function that uses the assert statements you created. Verify that everything runs as expected.

Name:

Spring 2024

Date:

CSCI 1300: Recitation 4

3 Valentine's Day

In a charming store named "MathCandies", a town surrounded by hills, the annual "Valentine Day" celebration is approaching. This store is renowned for blending mathematics with the art of candy design, creating unique candies adorned with mathematical designs.

As the mastermind behind this exciting venture, you're tasked with ensuring the success of Valentine Day by precisely calculating the ingredients for your candies. To achieve this, you'll employ two specialized functions to accurately calculate volumes of the heart-shaped candies the spherical candies. Whether it's employing the equation for your delectable hearts milk chocolate or the formula for your mouthwatering chocolate mousse truffle sphere, your creations will not only exhibit mathematical precision but also offer irresistible flavors.

- Equation for heart chocolate volume: $Volume = Area * Height = (\frac{1}{2}\pi r^2 + \frac{1}{2}\pi r^2 + \frac{1}{2}(2\sqrt{3}r \times 4r)) \times r = (\pi + 4\sqrt{3}) * r^3$
- Equation for mousse truffle sphere volume: $Volume = (\frac{4}{3}) * \pi * r^3$

```
/**
 * @brief Function to determine the volume of a heart-shaped candy using a radius
 * @param radius of the semicircles of heart-shaped candy
 * @return double - volume of heart-shaped candy
 */
double calculateVolumeOfHeartCandy(double radius)
{
    // Your code goes here.
}

/**
 * @brief Function to determine the volume of a spherical candy using its radius
 * @param radius of the spherical candy
 * @return double - volume of spherical candy
 */
double calculateVolumeOfSphericalCandy(double radius)
{
    // Your code goes here.
}
```

Problem 3.1. Write out the steps you would use to solve this problem by hand as pseudocode.

Heart

take parameter value
calculate volume using radius
save in variable
return variable as double

Sphere

take parameter value
do calculations of volume
save in variable
return variable

CSCI 1300: Recitation 4

Spring 2024

em 3.2. Pick two possible inputs for each of your two functions (four total). Follow the steps you for these values to find your result, and verify it.

$$\text{input} = 1$$

$$= \frac{\pi}{2} + \frac{\pi}{2} + \frac{1}{2}(2\sqrt{3} \cdot 4)$$

ve in variable

$$\text{input} = 2$$

$$(2\pi + 2\pi + \frac{1}{2}(2\sqrt{3} \cdot 2 \cdot 8)) \cdot 2$$

Save in Volume

$$\text{input} = 1$$

$$\frac{4}{3}\pi$$

Save in Volume

$$\text{input} = 2$$

$$\frac{4}{3}\pi \cdot 4$$

Save in Volume

Problem 3.3. Translate your inputs and expected outputs into assert statements.

$$\text{assert}(\text{calculate volume of heart candy}(2) == 2\pi + \frac{1}{2}(2\sqrt{3} \cdot 2 \cdot 8))$$

$$\text{rt}(\text{calculate volume of heart candy}(1)) == \frac{\pi(2\sqrt{3} \cdot 4)}{2}$$

$$(2\sqrt{3} \cdot 2 \cdot 8)$$

Problem 3.4. Translate your pseudocode into a c++ program to solve the above code, using your assert statements in your main function to verify that your program works as expected.

Submission Instructions: Create a zip file that contains your solution .cpp file for question 2, question three, and photos of this handout and submit on Canvas.