CSE 108 (January 2024)

Offline 01

Consider the following **main()** function definition of a C++ program. This function uses objects of three classes named **Rectangle**, **Triangle**, and **Circle**, along with a **ShapeCollection** class. You are required to write the full C++ program without changing the definition of the **main()** function. You can get hints about the member variables and the member functions of the classes from the code inside the main() function and from the expected output. Carefully read the comments inside the main() function to understand what needs to be done. In your program, you can only use two string functions: **strlen** and **strcpy** for handling dynamic memory in **char*** variables with malloc.

```
int main() {
    // Create rectangle with length, width, color
    Rectangle r1(10, 20, "Red");
    // The color is stored using malloc, which will be freed during destruction
    cout << "Rectangle Perimeter: " << r1.getPerimeter() << endl;
    cout << "Rectangle Area: " << r1.getArea() << endl;
    cout << "Rectangle Color: " << r1.getColor() << endl;
    // When changing the color, you need to free the memory of the old color
    // and allocate new memory for the new color
    r1.setColor("Yellow");
    cout << "Rectangle Color: " << r1.getColor() << endl;
    cout << "-------------------------------------" << endl;

    // Create triangle with a, b, c, color. (a, b, c are lengths of the sides)
    Triangle t1(3, 4, 5, "Blue");
    cout << "Triangle Perimeter: " << t1.getPerimeter() << endl;
    cout << "Triangle Color: " << t1.getColor() << endl;
    cout << "Triangle Area: " << t1.getArea() << endl;
    t1.setColor("Orange");
    cout << "Triangle Color: " << t1.getColor() << endl;
    cout << "-------------------------------------" << endl;

    // Create circle with radius, color
    Circle c1(7, "Green");
    cout << "Circle Perimeter: " << c1.getPerimeter() << endl;
    cout << "Circle Area: " << c1.getArea() << endl;
    cout << "Circle Color: " << c1.getColor() << endl;
    c1.setColor("Purple");
    cout << "Circle Color: " << c1.getColor() << endl;
    cout << "-------------------------------------" << endl;

    // Test ShapeCollection
    ShapeCollection shapes;
```

```
    // IMPORTANT: See notes [1], [2]
    shapes.addRectangle(r1);
    shapes.addTriangle(t1);
    shapes.addCircle(c1);

    Rectangle r2(15, 25, "Black");
    shapes.addRectangle(r2);
    Triangle t2(5, 12, 13, "White");
    shapes.addTriangle(t2);

    cout << "Number of Rectangles: " << shapes.getRectCount() << endl;
    cout << "Number of Triangles: " << shapes.getTriCount() << endl;
    cout << "Number of Circles: " << shapes.getCircCount() << endl;
    cout << "-------------------------------------" << endl;

    shapes.printRectangles();
    shapes.printTriangles();
    shapes.printCircles();

    return 0;
}
```

Expected output:

```
Rectangle Perimeter: 60
Rectangle Area: 200
Rectangle Color: Red
Rectangle Color: Yellow
-------------------------------------
Triangle Perimeter: 12
Triangle Color: Blue
Triangle Area: 6
Triangle Color: Orange
-------------------------------------
Circle Perimeter: 43
Circle Area: 153.86
Circle Color: Green
Circle Color: Purple
-------------------------------------
Number of Rectangles: 2
Number of Triangles: 2
Number of Circles: 1
-------------------------------------
Rectangle 0: length: 10 width: 20
Rectangle 1: length: 15 width: 25
Triangle 0: a: 3 b: 4 c: 5
Triangle 1: a: 5 b: 12 c: 13
Circle 0: radius: 7
```

You can copy the main() function shown above for your convenience: https://pastebin.com/Eg85nVWx

Submission guidelines:

1. Create a folder named after your student ID.
2. Write your code in a single file named your_ID.cpp.
3. Move the .cpp file into the folder from step 1.
4. Zip the folder and name it your student ID. Submit this zip on Moodle.

For example,
1805010/
└── 1805010.cpp
You would then submit 1805010.zip

Important Note:

If you need to access or modify any member variables of the classes, use appropriate getters and setters. Do not keep the member variables public.

The colors of the objects (rectangles, triangles and circles) should be stored using dynamic memory allocation. Specifically:

- You have to allocate memory in the object constructors for storing the color only. No need to allocate memory for anything else.
- You have to free the old color and allocate memory for new color in the setColor() method
- You have to free the color in the destructor.

Apart from these few places, you do not have to dynamically allocate or free memory anywhere else in your code.
You can perform dynamic allocation with either malloc / new (whichever you prefer). But make sure to do it correctly.

[1]: During construction of the ShapeCollection class, you **do not** need to **dynamically allocate** memory. Use **static arrays** to store up to **100 shapes of each type** (rectangles, triangles, and circles). Since the static arrays will be automatically freed, you don't have to handle them in the destructor.

[2]: You need to **pass the objects by reference** to the add functions. If you pass by value, the copy constructor will be called and the dynamically allocated char* will be copied, leading to double free errors (once when the parameter goes out of scope, and another when the original object is freed). Once passed by reference, do not directly store the reference in the array element. Instead, extract the data from the original object and set the attributes manually. Use appropriate getters and setters here.

More on pass by reference: https://www.ibm.com/docs/en/zos/2.4.0?topic=calls-pass-by-reference-c-only

Suggestion:
If you find it difficult to write multiple classes at once, you can do it stepwise. In the first step, implement the Rectangle class and test it individually with the corresponding part of the main() function. Once it works, proceed with implementing the Triangle and Circle classes. Finally, implement the ShapeCollection class to manage the collection of shapes and ensure all shapes are handled properly.

Deadline: September 15 (11:55 PM)

Please do not copy from any source (friends, internet, AI tools like ChatGPT, etc.). Doing so may lead to severe penalties.