# Module 7:
# Structures and Classes

COP2274

In-class Assignments

# **M7A** Class called Cone

1. Write **a class** called **Cone** which contains two *private member variables* (double) that represent the radius and height.

   - Create *a public accessor and a mutator* for each *private member variable*

2. Write *a public member function* of **Cone** called *surfaceArea()* which returns the surface area of the **Cone**.
3. Write *a public member function* of **Cone** called *volume()* which returns the volume of the **Cone**.

# **M7A** Class called Cone

4. In your main(), create a **Cone** object and prompt the user for the dimensions of a cone as shown in the test case. Use the mutators to set the user-inputted values and the accessors to get them. Call *surfaceArea() and volume()* on your **Cone** object to display the calculated surface area and volume as shown in the test case.

*Notes: Use the following formulas to calculate the surface area and the volume of a cone:*

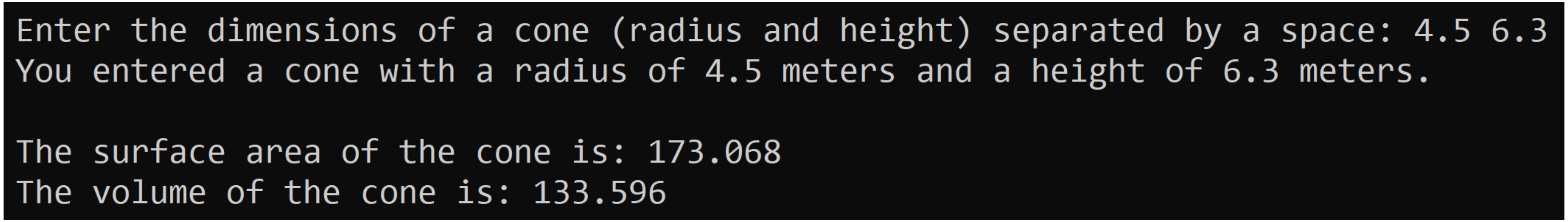
$$Surface\ Area = \pi r(r + \sqrt{h^2 + r^2})$$

$$Volume = \pi r^2 \frac{h}{3}$$

$$\text{where } r = base\ radius\ of\ a\ cone, h = height\ of\ a\ cone, \pi = 3.14159$$

# **M7A** Class called Cone

*Test case*



```
Enter the dimensions of a cone (radius and height) separated by a space: 4.5 6.3
You entered a cone with a radius of 4.5 meters and a height of 6.3 meters.

The surface area of the cone is: 173.068
The volume of the cone is: 133.596
```

# **M7B** Classes called Client and Bank

1. Write **a class** called **Client** that contains two *private member variables for the amount of money in checking, and in savings*.

   - Create *a public mutator* for the *private member variables*

2. Write *a public member function* of **Client** called *showData()* which displays the amount of money in checking and savings as shown in the test case.

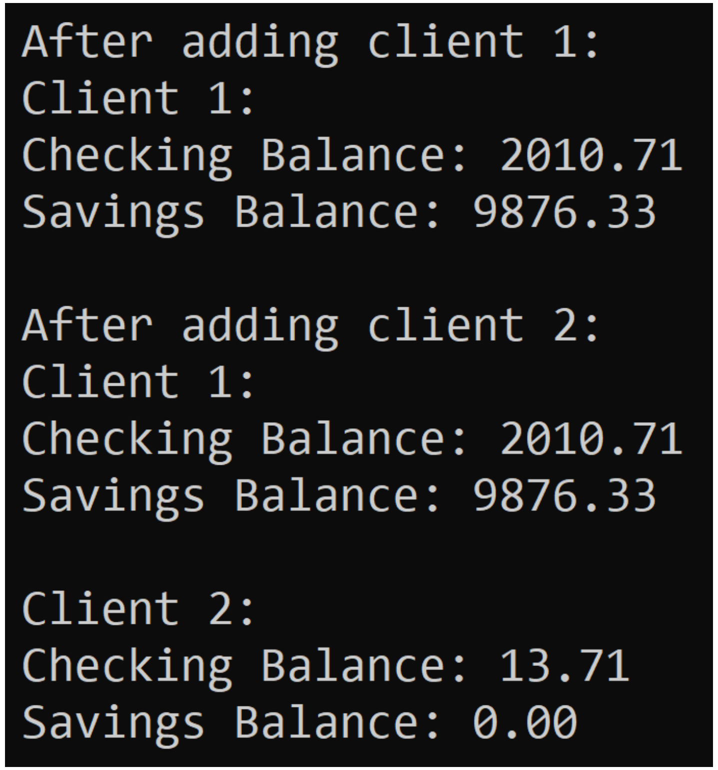
3. Write **a class** called **Bank** that contains three *private member variables for an array of the clients (**Client**),* the number of clients in the array (integer), and a capacity for that array (assume capacity is 3).

# **M7B** Classes called Client and Bank

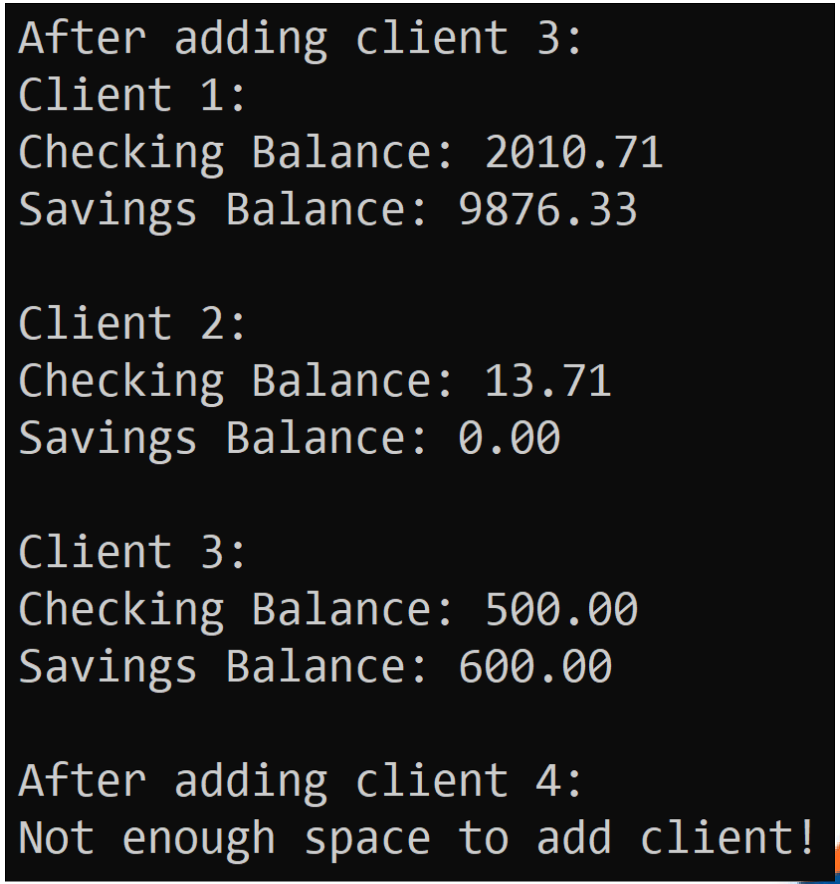
4. Write a *public member function* of **Bank** called *addClient()* which takes in an **Client** object and puts it in the internal array. It should increment the member variable storing the number of clients. If the array is already full (i.e. the size == capacity), *addClient()* should print an error message and DO NOT add the client to the array.
5. Also, write *a public member function* of **Bank** called *showData()* that simply calls *showData()* on all the clients in the array.
6. In your main(), test your **Client** and **Bank** classes and their member functions with some hardcoded (not from the user) values as shown in the test case.

# **M7B** Classes called Client and Bank

*Test case*



```
After adding client 1:
Client 1:
Checking Balance: 2010.71
Savings Balance: 9876.33

After adding client 2:
Client 1:
Checking Balance: 2010.71
Savings Balance: 9876.33

Client 2:
Checking Balance: 13.71
Savings Balance: 0.00
```



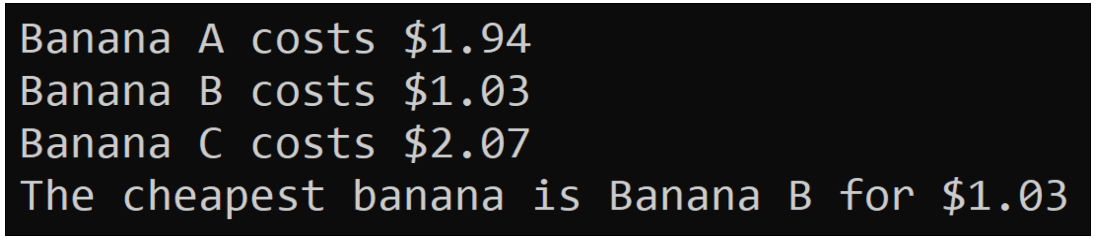
```
After adding client 3:
Client 1:
Checking Balance: 2010.71
Savings Balance: 9876.33

Client 2:
Checking Balance: 13.71
Savings Balance: 0.00

Client 3:
Checking Balance: 500.00
Savings Balance: 600.00

After adding client 4:
Not enough space to add client!
```

# **M7C** Class called Bananas

1. Write **a class** called **Bananas** which contains two *private member variables for the ID*(char) and price(double). Create *a public accessor and a mutator* for each *private member variable.*
2. Write a non-member function outside of the class **Bananas** called *cheapestBanana()* which takes in an array of **Bananas** objects (**Bananas**) and its size as parameters and returns a **Bananas** object that costs the cheapest price.
3. In your main(), test your **Bananas** class and non-member function *cheapestBanana()* after declaring an array of **Bananas** objects, initializing the array with hardcoded values, and display each element of the array as shown in the test case.

# **M7C** Class called Bananas

*Test case*



```
Banana A costs $1.94
Banana B costs $1.03
Banana C costs $2.07
The cheapest banana is Banana B for $1.03
```