Week 10

Intro to Computer Science - CSI 1430

Hello and welcome to the weekly resources for CSI 1430: Introduction to Computer Science! This week's resource will be covering the course material that will be covered in week 10 of the course.

The semester is ⅔'s through and as we start approaching finals try to go back an incrementally look over previous weeks material. This will not only prepare you for finals, but also strengthen your current C++ foundation. **Just as a reminder:** These documents are intended for going over concepts that are discovered in class, as they are a concise and to the point review of the materials. But remember, these documents are not a replacement for lecture and they cannot cover everything, so make sure you get help on any specific concepts you struggle with as well as go over lecture notes, programs, and the textbook.

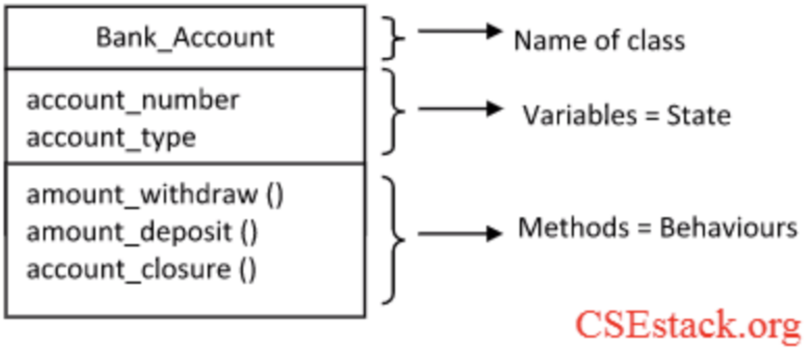
Reminder: If you have any questions about these study guides, group tutoring sessions, private 30 minutes tutoring appointments, the Baylor Tutoring Youtube channel or any tutoring services we offer, please visit our website https://baylor.edu/tutoring or call our drop in center during open business hours. Monday-Thursday 9am - 8pm on class days (254) 710-4135.

Keywords: *Classes*

TOPIC OF THE WEEK

Classes

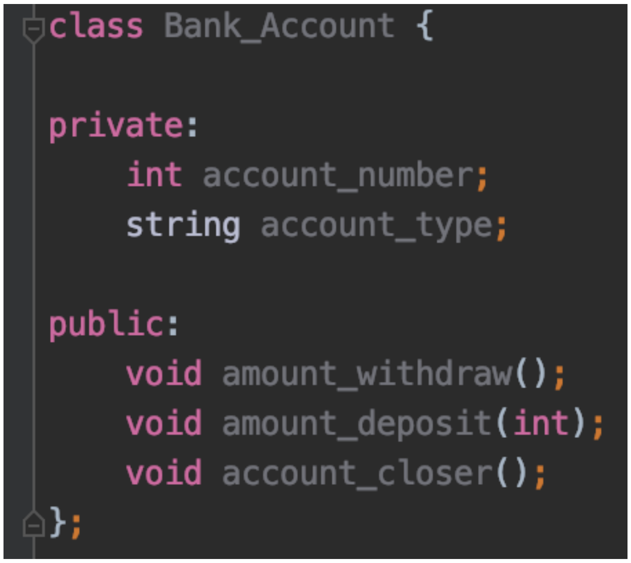
So far in this class, we have been writing code fairly linearly. This means that we are given some problem or task to accomplish and we have produced results that follow a sequence of instructions. Our first deviation from this came about when we introduced the concept of functions, where we were able to break apart our problems into subproblems to organize our code. This is known as functional-style programming. However, our programs will continue to get larger and the concepts contained within them will get continually more elaborate. These "concepts" are common day items or things such as a Person or a Car. In order to capture these concepts, we are going to invoke the use of classes. Classes are part of a programming model called object oriented programming, or OOP, that groups a certain number of attributes (or variables) under a single object name type. We can also define certain actions for these class objects with the use of functions that allow us to define behavior of the class itself.



Let's take a bank account for example to better define classes. Conceptually, the way we want to think about a class is in 3 main parts.

- Class name
- Class variables (or attributes)
- Class Methods

The class name is what the object we are creating is going to be referenced by. This is a pretty simple concept, as the object we are defining will normally have a name attached with it. In our example, the Bank Account class will be referenced by the name Bank_Account. Next is the class attributes, which show the different types of data that we are wanting to store within the class. These data types normally include the primitive variables available inside of C++, as well as other objects. All these attributes are included to define some "part" of the overall class. In our example, a bank account has a type and a number that we would like to package together under a single named object. This means that each instantiation on the Bank_Account class as an object will allow the storing of these types of data. Lastly, we have the methods. These methods define the behavior for the class. These are able to not only define some "inner functionality" for a class, but allow us to interact with the class in a safe way. We will dive deeper into what these functions can do for us in later weeks!



```
class Bank_Account {

private:
    int account_number;
    string account_type;

public:
    void amount_withdraw();
    void amount_deposit(int);
    void account_closer();
};
```

For now, take a look at how something like this would be visualized in code.

CHECK YOUR LEARNING

1. What does OOP stand for?
2. What are the 3 main parts of a class?
3. What is an attribute in a class responsible for?

THINGS STUDENTS MAY STRUGGLE WITH

1. Classes are not the only way we can achieve object oriented programming within C++. There is also the concept of a struct which we will explore in detail in later weeks. Take some time to look at what the differences and uses for both are in the meantime.
2. Once we initialize an object, we are able to set the different attributes individually! Either using what we will come to know as a setter/getter function or simply mentioning which part of the data we are accessing, we are able to retrieve and modify that data. The way we access this data all depends on what type of access it is given which will be explored later! The three types of access are:
    - Private
    - Public
    - Protected

Thanks for checking out these weekly resources! Don't forget to check out our website for group tutoring times, video tutorials and lots of other course resources: https://baylor.edu/tutoring. The answers to the 'Check Your Learning' questions are below.

Answers:

1. Object oriented programming
2. Name, attributes (or variables), and methods
3. An attribute is a specific variable for a function that is responsible for storing a specific part of an object's data.

Note: All tables were taken from the Baylor CS 1430 zyBook