**Task 1**

1. Pointers are used because it’s much easier to provide an address of an instance instead of providing the full instance this saves the need of using “getters” as by obtaining the address and accessing it, the information inside it are stored in “real-time” and everything that is needed was performed by the instance to the given address.
2. We can tell that Menu is an abstract class because it’s mainly using pointers and references and that’s because you can’t create an object of an abstract class type. Furthermore, “virtual” keyword makes it abstract as if class contains at least one virtual function it is considered as abstract class. Store could be possibly an abstract class, because the user itself won’t need to access the “full-store” at once, but he will be interested in particular sections of it. Furthermore, Application could be an abstract class as it only provides information about login’s and logout’s. **//needs more work**
3. The Accounts are not set to work with different types of users currently as there’s no actual logic to perform such action, what I mean by that is that function “login” simply sets boolean to true and that’s all, there’s no database or any kind of way which would allow us to store user information. In-order to make this work, we would need to find a way to store information in some sort of a database perhaps vector of strings. As this would allow us to distinguish Admin’s and User’s credentials.
4. Menu system is setup to work polymorphically in the following way:

* There are two instances of Option declared
* Each of them expects first parameter to be of different type
* Correct one is initated when particular type was passed in

Virtual functions are used to support “polymorphism” at run time, for instance if the virtual function is called by the base class then the compiler decides at runtime which method to use. Virtual functions are also overriding the behaviour for the base class, so it tells compiler which version to use instead of the default one. If virtual functions were not there then simply menu would be called in with default values and the aspect of polymorphism would be forgotten as there was no actual “path” to lead to the desired function.

1. It’s good idea to make Date::CurrentDate() a static function because when a static member is declared it clearly means that no matter how many instances of that class are created, there will be only one copy of the static member. Additionally, CurrentDate will be same doesn’t matter how many times you launch the program the date will be same. Furthermore, static members are shared by all the object of the class, so it’s wise idea to have CurrentDate as static, because it will be shared across different functions and there will be only one date. As well static member functions can be called even if nothing was craeted within the class and lastly static functions can only be accessed by using the class name and scope resolution operator(double colon, : ).