**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