HW2- DSA

1. Yes, C++ allows you to specify more than one definition for a function name or an operator in the same scope, which is called function overloading and operator overloading respectively.
2. Yes, abstract classes act as expressions of general concepts from which more specific classes can be derived. You cannot create an object of an abstract class type; however, you can use pointers and references to abstract class types. A class that contains at least one pure virtual function is considered an abstract class. Classes derived from the abstract class must implement the pure virtual function or they, too, are abstract classes.
3. The main difference is that abstract classes (run-time polymorphism) are a run-time mechanism, while templates are a compile-time mechanism. This means that by using abstract classes you can possibly change the behaviour at run-time. With templates, instead, you bind your implementation at compile-time. Being a run-time mechanism, abstract classes bring run-time overhead due to virtual functions. Templates don't suffer this issue. Also, an abstract class instead is actually a "specification" of the concrete classes you can use.
4. Yes, we can define class members static using static keyword. When we declare a member of a class as static it means no matter how many objects of the class are created, there is only one copy of the static member.