OOP Assignment 2

Group Report

Group UT1

Chad Simpson, Luke Ward, Shaun Webb

***Question 1 :***

**a)**

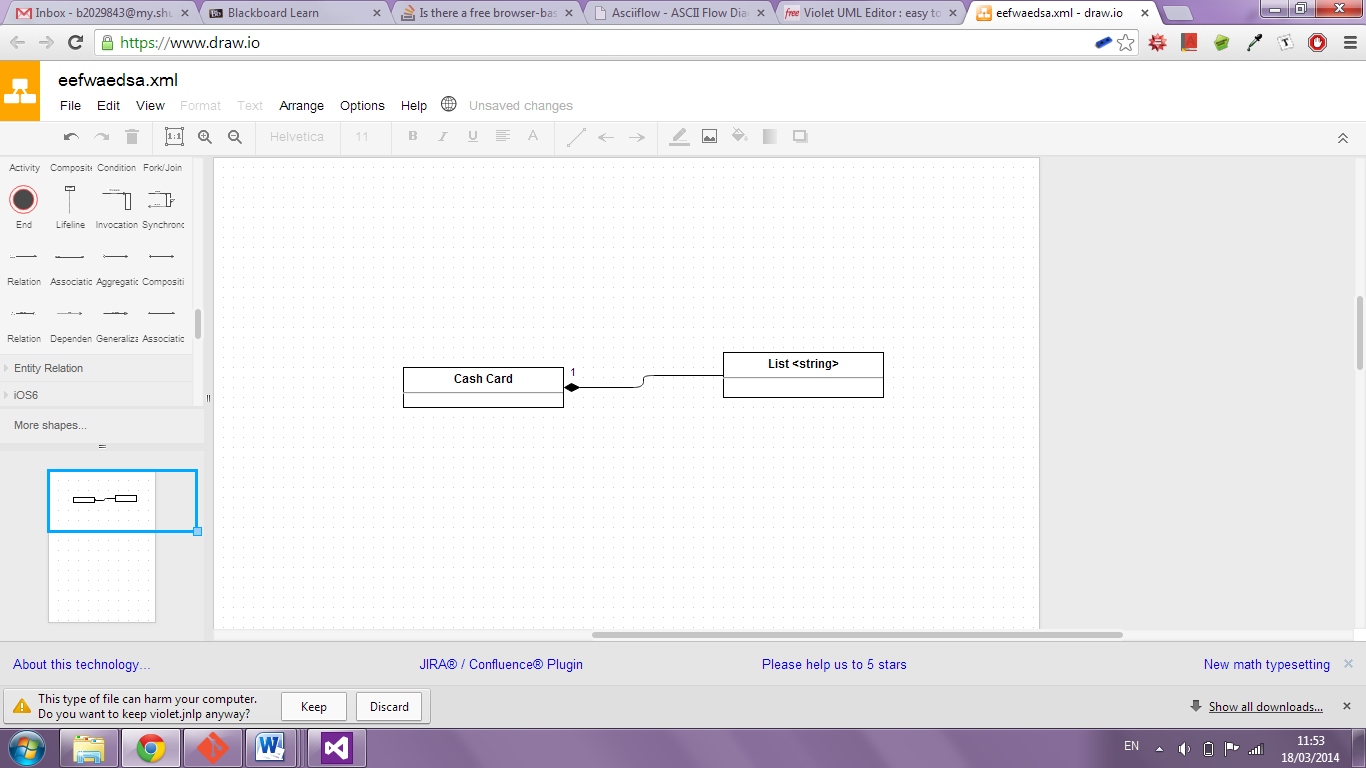
It is possible for the canWithdraw function to be inherited by the ChildAccount class, due to the fact that it is a child of the SavingsAccount class, which is a chile of the BankAccount class.

One of the possible problems that would arise with using the canWithdraw function in the ChildAccount class is that it has a minimumBalance due to the fact that it is a child of the SavingsAccount. The BankAccount version of the canWithdraw function will not be taking into account the minimumBalance of the SavingsAccount classes which could then have a possibility of withdrawing below the minimumAmount.

**b)**

**c)**

The relationship between the CashCard class and the List<string> classes is composition. The C++ mechanism that is involved in the implementation of it is that the List<string> is declared inside of the CashCard class.



**d)**

The UserInterface class is not an abstract class; this is due to the lack of pure virtual functions within the class. Abstract classes are useful as base classes only, and must have at least one pure virtual function present.

The only situation in which it would be appropriate for UserInterface to be an abstract class is if the end product needed different GUIs for different platforms, such as mobile and PC. In this case, you would need UserInterface as a base abstract class which the other UI classes could use as a parent class.

For the purpose of this program however, the only use will be on the ATM platform exclusively, negating the need for having UserInterface as an abstract class.

**e)**

In this program the function Date::currentDate() is required to be accessed outside of its class. Declaring the function as static allows currentDate to be called without using a class member object. In this program this is useful as it allows the current date to be accessed without creating a new Date object.