**Assignment 3**

Group project (team of 2 people)

Create Account class. It must contain name, *balance, dollars, euros, tenge* attributes with default name “Unnamed Account” and default balances of 0. Class must contain deposit and withdraw methods. Provide user define constructors which initializes attributes.

1. Create class SavingsAccount, which inherits from Account. It must contain interest rate as attribute and provide default value for interest rate. Every time when money is deposit the amount is multiplied by (1+interest\_rate/100) (with different interest rate for different currencies) (15%)
2. Add CheckingAccount class to the hierarchy. Checking account has name, *balances* and a fee of 1.5% (default). Every withdraw is charged with that fee. (15%)
3. Add TrustAccount to the hierarchy it must have name, *balances* and interest rate. The deposit method works same as SavingsAccount. Any deposit of 1000dollar or more gets a 10 dollar bonus (default). The amount that is allowed to be withdraw must be smaller than 20% of the balance. (15%)

Task 2.

1. Modify Account class so it is Abstract class by adding following pure virtual functions: (15%)

* virtual bool deposit(…) = 0;
* virtual bool withdraw(…) = 0;

1. Create I\_Printable class interface and provide functionality so all accounts are printable (display total balance and balances in *dollars, euros, tenge*) (15%)

Task 3.

1. Create Redistribution class with redistribute(double **tenge**) method, and every time the redistribute method is called the amount in **tenge** get distributed to All objects of Account related classes.

**Tip:** you may group objects in array or vector. (+15%)

1. Test the code with base class pointers as well as local objects. (Show screenshots) (10%)

=====

1. Bonus: work with git and upload files to Github, have at least 3 commits. Provide link (+2%)
2. Bonus: In task 2 part 2: Create I\_Printable class interface and provide functionality so all accounts are printable **to an ostream using overloaded insertional operator.** (+5%)
3. Penalty: similarity >70% (-100%)
4. Penalty: late submission (-5% for each hour)

* Advanced: import data about currency rates (using data parsing or external API) (+10%)