The objective of this lab is to

- DB handling (select)
- Classes

Instructions!

- Keep your student identity cards with you.
- This is an individual lab, you are strictly **NOT** allowed to discuss your solutions with your fellow colleagues, even not allowed to ask how he/she is doing, it may result in a negative marking.
- You can **ONLY** discuss this with TAs or Ma'am.
- Save your work frequently. Make a habit of pressing CTRL+S after every line of code you write.
- This is a **GRADED** lab, so, at the end of the lab session, you should have your complete work ready for evaluation.
- Follow proper coding conventions and write comments.
- ***Total Time for this Lab is 90 minutes.***

Task 01 [10 Marks]

ATM System in python

Before starting the task, kindly run the following create table statements in your SQL server and insert default data in it.

```
create table users (
      user_id int Primary Key auto_increment,
      account_no varchar (250) UNIQUE,
      account_balance int,
      password varchar (50)
);
```

In this task, you will implement your previous lab using classes. Create following classes

User class

- **Private Members** - Account number, password and account balance
- getter and setters to access private members
- Method to select data from database table(users) and display it

ATM class

- Inherit User class to access their data members
- Implement following functions
  - **register_account** A User can register his/her account, with account number and password.
  - **Login** Once a user is registered, one can login with his/her account number and password. After login, user can check his/her balance, withdraw and deposit

Task 02 [10 Marks]

In this task, you will create a Time class and initialize it with following private members

- **hours**
- **minutes**
- **seconds**

Implement following methods:

- **getter and setter** with appropriate access modifiers
- **add_time** which should take two-time object and add them.
- **display_time24 and display_time12** which should print the time in 24 and 12-hour format
- **display_minute** which should display the total minutes in the Time
- **update_time** take minutes from user and convert it into appropriate time