

**ECE 220 – Computer Programming for Engineering – Winter 2020****Laboratory/Assignment: “OO...”  
Lab/Assignment****Objective**

The goal of this lab/assignment is to make you familiar with principles of object-oriented programming. In particular, you will be asked to perform a number of small programming exercises that allow you to learn basics of creating classes and their methods, creating instances, as well as be exposed to the concept of inheritance.

**IMPORTANT:** This lab should be done individually.

**Development Environment**

**IMPORTANT:** the whole lab/assignment should be done on the website called onlineGDB (<https://www.onlinegdb.com>). Both classes `BankAccount` and `saveBankAccount` should be in one file.

**Submission**

The submission of this lab should be done via eClass. Copy and paste your work from [www.onlinegdb.com](https://www.onlinegdb.com). Your code **HAS TO WORK** (be compiled and executed) on this website. The due date is Friday, **April 10<sup>th</sup>, 6:00 PM**. The penalty is 10% for each day of delay, and the final day of submission is Friday, April 17<sup>h</sup> at 6:00 PM.

**Problem Specification**

In this lab/assignment, you use a mechanism of inheritance to design and implement a class. The activities of this lab are very much related and similar to the tasks of the pre-lab. Therefore, a successful completing of the pre-lab is essential for this lab.

**Individual tasks:**

1. based on `BankAccount` class from the TUTORIAL create (via inheritance) a new class `saveBankAccount` with additional three members:  
    `double interestRate,`  
    `int noWithDraws,`  
    `int MAXnoWithDraws,`  
and a constructor that accepts:  
    `name, account_number, balance, interestRate, and`  
    `MAXnoWithDraws, and assigns 0 to noWithDraws`

2. a new version of the method `printSummary()` that allows to print also (besides `name`, `account_number`, `balance`) the values of `interestRate`, `noWithdraws`, `MAXnoWithdraws`
3. in the main program create an object:  
    `tomAcc` as an instance of `saveBankAccount` with: `name:Tom`,  
    `account_number:234567`, `balance:2000`, `interestRate:0.05`,  
    `MAXnoWithdraws:1`  
    display its content using `printSummary()`
4. create two new methods for the class `saveBankAccount`:  
    `callInterest()`  
    it calculates interest, add it to the balance, and display new balance  
    `void resetWithdraws()`  
    it resets (sets to 0) the member/field `noWithdraws`  
and a new version of the method  
    `double withdraw(double)`  
    it checks and updates `noWithdraws`, checks if the operation can take place, takes the amount to withdraw as `double`,  
    display new balance, and returns actual amount of withdraw in `double`
5. in the main program, perform the following operation on the object `tomAcc` (created in point 3)  
    deposit 1000 to Tom's account  
    withdraw 500 form Tom's account  
    withdraw 500 from Tom's account one more time  
    display balance of Tom's account  
    reset `noWithdraws` of Tom's account  
    withdraw 500 from Tom's account one more time  
    display balance of Tom's account  
    call function/method to calculate interest  
    display balance of Tom's account

**Marking Scheme**

This assignment is worth 6% of your final mark. A total number of points you can obtain is 100. The marking of the lab is done according to the following schema:

<b>TASK</b>	<b>POINTS</b>
Building an inherited class with additional methods and a constructor	/30
Modifications of the method <code>printSummary()</code>	/25
Writing new methods	/30
Writing the main program and usage of the methods	/15
<b>Total</b>	<b>/100 points</b>