

Data structure
Lab 2 – class and object
Dr. Sophea PRUM
sopheaprum@gmail.com

If a programmer codes just for fun he has all his skill.
If he codes for score his hand tremble and his breath is uneasy

Preparation:

- Lunch your NetBeans IDE and create a project called “LabPratice2”. All the exercises must be created under this new project.

Note: from exercise 1 to 4, try to answer the questions before implement the given code. **Be a detector!** It is fun. Then, implement the given code and check your answers!

Exercise 1: Observing the result

Given the class bellow:

```
public class IdentifyMyParts {
    public static int x = 7;
    public int y = 3;

    public static void main(String[] args) {
        IdentifyMyParts a = new IdentifyMyParts();
        IdentifyMyParts b = new IdentifyMyParts();
        System.out.println("Before modifying the values");
        System.out.println("a.y = " + a.y);
        System.out.println("b.y = " + b.y);
        System.out.println("a.x = " + a.x);
        System.out.println("b.x = " + b.x);
        System.out.println("After modifying the values");
        a.y = 5;
        b.y = 6;
        a.x = 1;
        b.x = 2;
        System.out.println("a.y = " + a.y);
        System.out.println("b.y = " + b.y);
        System.out.println("a.x = " + a.x);
        System.out.println("b.x = " + b.x);
    }
}
```

Q1: what is (are) the class field?

Q2: what is (are) the instance field?

Q2: what are the output of this main method (see the code below)? Explain why the values of *a.x* and *b.x* are still the same after modifying their values?

Exercise 2: SomethingIsWrong

Q1. What's wrong with the following program?

```
public class SomethingIsWrong {
    public static void main(String[] args) {
        Rectangle myRect = null;
        myRect.width = 40;
        myRect.height = 50;
        System.out.println("myRect's area is " + myRect.getCenterX());
    }
}
```

Exercise 3: worker

Q: what is the output of the code below?

```
class Worker {
    String name;
    int age;
    double wage;

    Worker(String name, int age, double wage) {
        this.name = name;
        this.age = age;
    }

    public static void main(String[] args){
        Worker mc = new Worker("Peter",25,235.0);
        Worker mc2 = new Worker("Alan",64,434.0);
        Worker mc3 = new Worker("Emily",36,320.0);
        System.out.print(mc3.age + ", " + mc2.wage + ", " + mc2.name);
    }
}
```

Exercise 4: PointCoordinates

```
public class PointCoordinates {
    private int x;
    private int y;

    public PointCoordinates(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public void translate(int dx, int dy){
        this.x = this.x + dx;
        this.y = this.y + dy;
    }
}
```

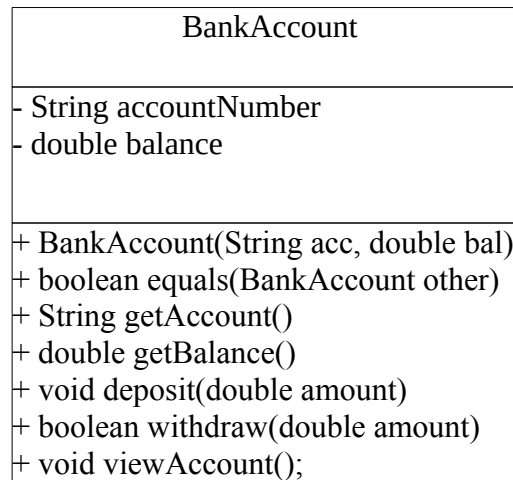
Consider a class `PointCoordinates` above:

Q: what is the output given by the main method below?

```
public static void main(String[] args) {  
    PointCoordinates p1 = new PointCoordinates(2,2);  
    PointCoordinates p2 = new PointCoordinates(5, 10);  
    PointCoordinates p3 = p2;  
    p3.translate(5,-2);  
    System.out.println("p1.x="+p1.getX());  
    System.out.println("p1.y="+p1.getY());  
    System.out.println("p2.x="+p2.getX());  
    System.out.println("p2.y="+p2.getY());  
    System.out.println("p3.x="+p3.getX());  
    System.out.println("p3.y="+p3.getY());  
}
```

Exercise 5: BankAccount

Now, you will learn how to create your first java class. The UML diagram of your class “BankAccount” is illustrated bellow:



This class “BankAccount” have

- 2 fields:
 - accountNumber: an identifier that never changes
 - balance: potentially does change
- 1 constructor
 - Each time an object of class “BankAccount” is created, the values of *accountNumber* and *balance* must be provided. Therefore, this constructor take 2 parameters as input (acc, and bal)
 - acc is account number
 - bal is balance
- 6 methods
 - **boolean equals(BankAccount other)**: a public method having one parameters *other* which represents other object of BankAccount. This method returns true if the accountNumber of the current object is the same as the accountNumber in the object *other*.
 - **String getAccount()**: a public method which returns the bank account number of this account
 - **double getBalance()**: a public method which returns the balance of this account
 - **void deposit(double amount)**: deposit money in the bank account. This method returns nothing
 - **boolean withdraw(double amount)**: a public method allowing to withdraw money from this account. The withdraw amount must be positive and not zero. Otherwise, the system return false. In addition, the method will refuse the withdraw operation (return false) if the withdraw amount is higher than the balance. If the operation is success, the method return true.
 - **void viewAccount()**: is public method allowing to view the accountNumber and its balance. This method return nothing. It simply prints out “*Account number xxxxxxxx having yyyy\$*”. Where xxxxxxxx must be replaced by the accountNumber and yyyy must be replaced by balance.

TODO1: In your project “LabPratice2”, create a java class and name it “BankAccount”. This java class should not contain main method. Implement your class based on the UML diagram and description given above.

TODO2: Create a new java class called “BankApplication”. This java class contains the main method.

- Q1: Now create an instance called myAccount having accountNumber = “112233445566” with 200\$ as the initial balance.
- Q2: Now you want to view your account
- Q3: You just earn 150\$ from your part-time job. Now you deposit all your money and view your account
- Q4: You are planing a trip to Penang. So you withdraw 300\$ from your account. The system will print-out “*successfully withdraw yyyy*” if withdraw operation is success, where yyyy is the withdraw amount. Otherwise the system will print-out “*withdraw operation has been refused*”.
- Q5: Now again, you want to buy some books, so you wish to withdraw another 100\$. The system will print-out “*successfully withdraw yyyy*” if withdraw operation is success. Otherwise the system will print-out “*withdraw operation has been refused*”.
- Q6: Create a new instance (bank account) called *otherAccount*. This account having accountNumber = “002233445566” with 500\$ as the initial balance. Now, view the otherAccount object
- Q7: We want to check if myAccount and otherAccount are different or the same. The system will print out “*Duplicate account number*” if this two account are equal. Otherwise, the system will print out “*The two account are different*”.