Integration of Object Oriented Programming Concepts

Let’s demonstrate the functionality of the expression **o.l⇐ ς(x)b** via Java syntax that is shown below.

```
public class BankAccount {

    public BankAccount ( double aBalance){

        currentBalance = aBalance;

    }

     ⋮

    public void deposit (double anAmount) {

        double tempBalance = anAmount + currentBalance;
        currentBalance = tempBalance;

    }

    public double getCurrentBalance(){

        return currentBalance;

}

    private double currentBalance;

}

public class BankAccountTester {

    public static void main(String args[]) {

         ⋮

    BankAccount anAccount = new BankAccount();
    anAccount.deposit(500);

    }

}
```

The Java code consists out two classes where the **BankAccount** class is a sub-class or a module and holds the code of the program and the main class **BankAccountTester** is the main class, which contains the method **main** that is responsible for testing the program for correctness and executing it. The sub-class connects to the main class by using the object such as **BankAccount anAccount = new BankAccount();** which matches the name of the sub-class **BankAccount** and its contents is stored in the instance field in the sub-class **private double balance** where **private** is the access specifier and allows modification to the variable balance through the method **deposit**. Its data type is a **double** and name is **currentBalance** can be classified as a declaration, since data type and name means exactly that. The line **anAccount.deposit(500)** is in the main class the equivalent to the OOC **o.l**.

The **public void deposit(double anAmount){ …}** and the code inside the parentheses is the equivalent of the expression **ς(x)b** method with parameter **x** and body **b**. The expression **o.l⇐ ς(x)b** operates on two classes, the sub-class **BankAccount** where **ς(x)b** is found and the main-class **BankAccountTester** where **o.l** operates and executes the code. Again, the connection between the sub-class and the main class is the object **BankAccount anAccount = new BankAccount();** where the **BankAccount** matches the name of the sub-class.

Self-Check Questions for The Integration of Object Oriented Programming Concepts Sub-Section

1. How does the sub-class connect to the main class?
2. What is the difference between a sub-class and a main class?