General Sir John Kotelawala Defence University

Department of Computer Science

Object Oriented Programming I

Lab Sheet 9 Polymorphism – Method Overload & Method Override

1. Create a class called **Calculator** that provides methods for performing mathematical operations with a varying number of operands:
    - Method called **'add'** to add **two integers** and return the sum
    - Method called **'add'** to add **three integers** and return the sum
    - Method called **'add'** to add an **array of integers** and return the sum

    - In the main method, create an instance of the Calculator class and use it to perform addition operations with different numbers of operands.
    - Depending on the number of arguments, calling the **add'** method.

- Hint - Java will automatically select the appropriate version of the method to execute

2. Create a class called Converter that provides methods for converting values between different data types:
    - Method called **'converter'** to convert double to int
    - Method called **'converter'** to convert int to double
    - Method called **'converter'** to convert String to int

    - In the main method, create an instance of the Converter class and use it to convert values between different data types.
    - Depending on the number of arguments, calling the **converter'** method

3. Create a base class called '**Departments'** with method **display(),** which is just printing the **"There are three departments".**
    - Create a child class called '**ComputerScience**' and inside the child class create a method called **display()** which is just print the '**Teaching Computer Science'** that override the parent class play() method
    - Create a child class called '**ComputerEngineering**' and inside the child class create a method called **display()** which is just print the **'Teaching Computer Engineering**' that override the parent class play() method

- Create a Main class called **TeachingModules** and create instances of **ComputerScience**' and **ComputerEngineering**' but store them in references of type Departments'.
    - (This demonstrates polymorphism, where objects of different subclasses can be treated as instances of their common superclass.)

4. Create a base class called '**BankTransaction'** with the method **describeTransaction (),** which is just printing the **"This is a generic bank transaction".**
    - Create a child class called '**DepositTransaction'** and inside the child class create a method called **describeTransaction ()** which just prints the '**This is a deposit transaction. Adding funds to your account.'** that overrides the parent class **describeTransaction** () method.
    - Create a child class called '**WithdrawTransaction** and inside the child class create a method called **describeTransaction ()** which just prints the **'This is a withdrawal transaction. Taking funds from your account. 'that** overrides the parent class **describeTransaction** () method
    - Create a Main class called '**BankingApplication'** and create instances of **BankTransaction', DepositTransaction', and ' WithdrawTransaction'** but store them in references of type **BankTransaction**.
    - Your output should like this.