# CP 215 Object Oriented Programming in Java

## PRACTICAL_4

*4.1 The objective of this practical is to enable student to declare and use methods, object, instance and local variables and constructors.*

1. Write a java class containing *main* method and another method called *display*. The method *display* shall be called within *main* method to output Hello CP 215 Class to the screen. Method *display* should not have parameters or return type.

2. Write java class containing *main* method and other two methods called *addition* and *myMedthod*. Method *addition* should display the sum of two integers passed through its parameters *a* and *b*. *myMethod* should display title *"Adding two Integers:"* which is passed through its string parameter called *title*. All the methods should be called within main method. Note: Arguments should be hardcoded or embedded with the code.

3. Modify question 2 to allow a user to pass arguments to the methods using Scanner class.

4. Write a java class containing three methods; *main*, *display,* and *addition* methods. Method *display* should be responsible for outputting results to the screen. Method *addition* should add two integers passed through its parameters *a* and *b* and **return** the sum to method *display.* Method *display* should be called by the *main* method.

5. Write a java class that contains two methods; *main* and *addition.* Method *addition* should be overloaded to perform addition of two integers, addition of two double, addition of three integers and concatenation of two strings. Results should be displayed in *main* method. All values should be supplied by user via keyboard.

6. Create a class called *Student* with four instance variables; *nam*e, *regNumber*, *yearOfStudy*, and *gender.* Create *set* and *get* methods for inserting and returning values from the instance variables respectively. Values to the variable should be hardcoded through the parentheses of the method during the call.

7. Modify question number 6 by separating it into two classes; Class *Student* and class *myMain.* Class student should have four instance variables; *nam*e, *regNumber*, *yearOfStudy*, and *gender.* While class *myMain* should be used to define main method, create object of class student and display values. Values to the variable should be supplied by the user and passed through the parentheses of the class's object.

8. Write java program(s) that illustrate declaration and usage of local variables, instance variables, instance methods, class variables and class methods. Use comments to locate local, instance, and class variables and methods declared.

9. Write java class that has one instance variable, a programmer defined constructor which initializes that instance variable of the class and one method that accepts an integer value. Write another java class that instantiates an object of the class that was created and calls the method defined in the class. Use comments to indicate a parameter and an argument.

10. (Employee Class) Create a class called Employee that includes three instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Provide a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, do not set its value. Write a test app named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display each object’s yearly salary. Then give each Employee a 10% raise and display each Employee’s yearly salary again.

11. (Invoice Class) Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables—a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write a test app named InvoiceTest that demonstrates class Invoice’s capabilities.