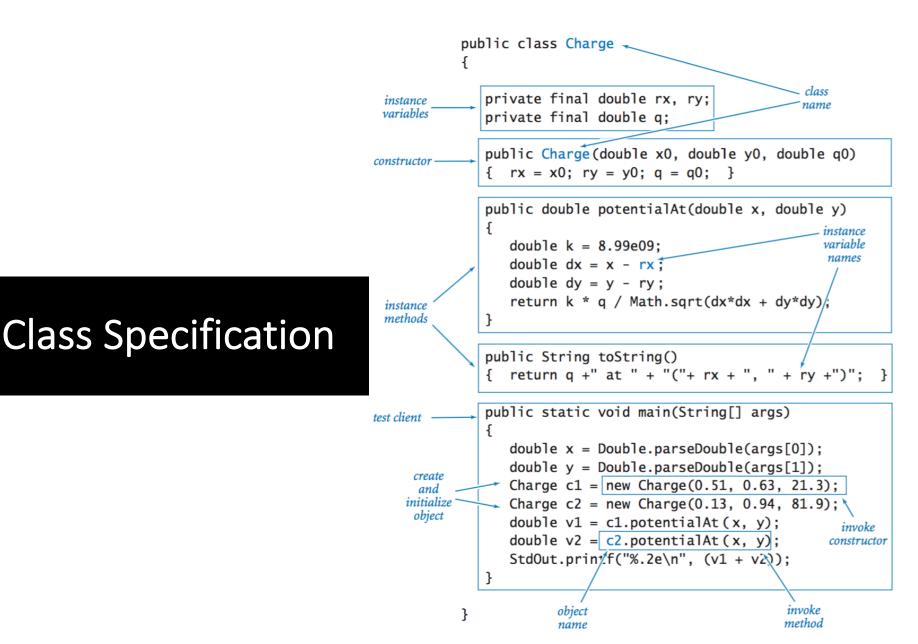


Java Programming

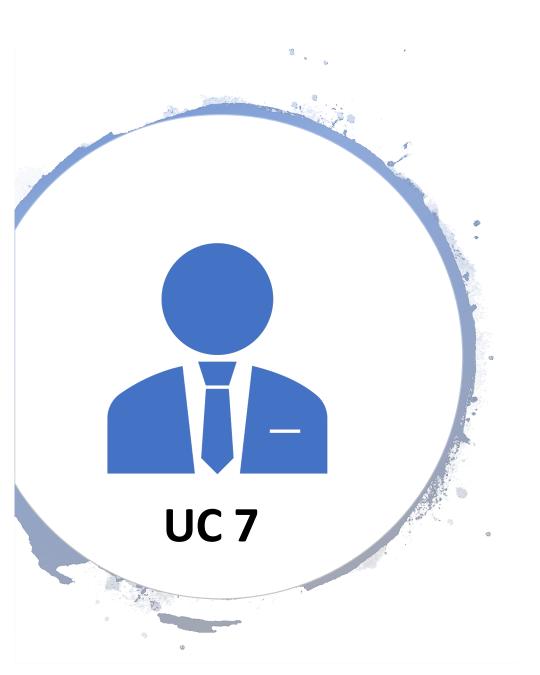
## Class,Objects &Methods

- A Class can be considered as a blueprint using which you can create as many objects as you like.
- Objects have state and behaviour
- Abstraction is a process where you show only "relevant" data and "hide" unnecessary details of an object from the user.
- Encapsulation simply means binding object state(fields) and behaviour (methods) together. If you are creating class, you are doing encapsulation.
- Association establishes relationships between two Objects so as to enable Method Invocation



BridgeLabz

Employability Delivered



Refactor the Code to write a Class Method to Compute Employee Wage

## Compute Employee Wage Using Class Methods

```
public class EmpWageBuilderClassMethod {
    public static final int IS_PART_TIME = 1;
    public static final int IS_FULL_TIME = 2;
    public static final int EMP_RATE_PER_HOUR = 20;
    public static final int NUM_OF_WORKING_DAYS = 2;
    public static final int MAX HRS IN MONTH = 10;
    public static int computeEmpWage() {
        // Variables
        int empHrs = 0, totalEmpHrs = 0, totalWorkingDays = 0;
        // Computation
        while (totalEmpHrs <= MAX_HRS_IN_MONTH &&
               totalWorkingDays < NUM_OF_WORKING_DAYS) {
            totalWorkingDays++;
            int empCheck = (int) Math.floor(Math.random() * 10) % 3;
            switch (empCheck) {
                case IS PART TIME:
                    empHrs = 4;
                    break;
                case IS_FULL_TIME:
                    empHrs = 8;
                    break:
                default:
                    empHrs = 0;
            totalEmpHrs += empHrs;
            System.out.println("Day#: " + totalWorkingDays + " Emp Hr: " +empHrs);
        int totalEmpWage = totalEmpHrs * EMP RATE PER HOUR;
        System.out.println("Total Emp Wage: " + totalEmpWage);
        return totalEmpWage;
    public static void main(String[] args) {
        computeEmpWage();
EmpWageBuilderClassMethod.java (END)
```



## Thank You