**Exercise – Basic Java**

lab1 - Object Oriented & Advanced Class Features

1. Write comp.Employee class with the following attributes:
   1. String name
   2. Double salary

Provide constructor that takes all the arguments and getters/setters methods

Override toString() to return full personal details

Override equals() to return true for two employees with the same name and salary

1. Write 3 Employee subclasses:
   1. Secretary – with additional attribute: String office
   2. Engineer – with additional attribute: String specialty
   3. Manager - with additional attribute: String department

For each subclass:

1. Override toString() & equals()
2. Provide constructor that takes full values
3. Add getters/setters
4. Write 1 Manager subclass:
   1. Director – with additional attribute: String group

For this subclass:

1. Override toString() & equals()
2. Provide constructor that takes full values
3. Add getters/setters
4. Create Test class with main method. Main should do the following:
   1. Create an array of Employees with size of 10
   2. Fill the array with 2 secretaries, 2 managers, 3 employees, 1 director and 2 engineers
   3. Print all employees details
   4. Print average salary
   5. Print management average salary

lab2 - Collections

1. Create class Company
   1. Code it to be a Singleton
   2. Add the following attributes:
      * List<Employee> employees
      * String name
   3. Provide a constructor that takes company name
   4. Provide getName() method
   5. Override toString() to print all employees details
   6. Add an ability to manage employees :
      * addEmployee(Employee)
      * removeEmployee(Employee)
      * getEmployees()
   7. Add report methods:
      * getAverageSalary()
      * getManagementAverageSalary()
      * getYearlyPayment()
      * getTotalNumOfEmployees()
      * getTotalNumOfManagers()
2. Update Test class to :
   1. Get a Company instance
   2. Print employees details
   3. Fill it with employees data
   4. Call all of its report methods

lab3 - Exceptions

1. Create an Exception class – CompanyReportException with the following attribute
   1. String companyName
   2. String reportName

Provide a constructor that takes all arguments including message (belongs to superclass..) and getter methods

1. Update Company report methods to throw CompanyReportException if the Employee list is null or empty. Each method specifies 'reportName' accordingly and uses "Data Unavailable" as exception message. Exception is also populated with company name
2. Now, fix Test class to handle the exception thrown from all report methods. When catching an exception – its full details are printed.
3. Run main without populating company with employees to cause an exception and test it.