**Home Assignment – 6  
Total Points: 100  
Due Date: 11/7/2018 (End of Day)**

***This assignment is based on topics covered in chapter 7 in the textbook.***

**Part - I**

**Type your answers to the following questions in a word document.**

1. What is the purpose of a constructor? Is there a limit on the number constructors we can have in a user defined class?
2. What is the rule in java with regards to the name of the constructor? Do constructors have return type?
3. What is the difference between parameters and arguments?
4. Identify the differences between local variable, instance variable and parameter variable
5. Differentiate between instance variable and static variable.
6. Explain why should instance variables be declared private ideally?
7. Explain the difference between accessor and mutator methods?
8. What is the difference between arguments and parameter variables.
9. Explain the purpose of toString() and equals( ) method.
10. When you are passing objects to methods, is it a good idea to assign the reference to the object directly to the instance variable? Explain your answer.

(Each question is 2 points each = 1.5 \* 10 = 15)

**Part - II**

**Work on the following exercises at the end of chapter 7:**

7.18.2 Reading and Understanding Code

#17, #18 and #27 (2 point each: 3 \* 3 = 9)

7.18.3 Fill in the code

#29, #31, #36 and #37 (3 points each: 2 \* 4 = 8)

7.18.4 Identify Errors in Code

#39, #46 (3 points each: 2 \* 2 = 4)

7.18.5 Debugging Area – Using Messages from the Java Compiler and Java JVM

#49, #52 (3 points each: 2 \* 2 = 4)

**Part – III: Programming Exercise (60 points)**

Design a User-Defined class called **Date.java** with three attributes day, month and year. A date object represents a date according to the Gregorian calendar. ***Please note you could copy and paste the code from Date.java that you created for your last lab assignment.***

In the same NetBeans project folder you created for the Date class, create another user-defined class called *Employee*. An employee object stores information about an employee. Each employee has a *employeeID*, first name, last name, birth date and hire date. You will implement the following the Employee class,

* Instance variables *employeeID, firstName, lastName, birthDate* (Date type), *hireDate* (Date type)
* Variable to keep track of the count of *Employee* objects created. Note there will be only one copy of this variable that is shared by all the objects of this class
* Create a static variable – nextID (initialized to 1000) to assign the next ID to the employeeID. There would be only one copy of this variable shared by all the objects of the class. This variable would be incremented each time a new Employee object is created.
* Create appropriate constructors. Note that to initialize *birthDate a*nd *hireDate* the constructors will need parameters of *Date* type. Also, you would not pass a value through parameter to initialize employeeID rather you would use the variable you created in earlier step (nextID) to initialize employeeID.
* Get and set methods for *firstName, lastName, birthdate and hireDate*
* Create a method *getEmployeeCount( )* that returns the current count of the Employee objects
* Create a *toString( )* and *equals* method ( )
* Include JavaDoc style comments along with pre-conditions, post-conditions for each of the mutator methods. The comments should also include a description of the method along with information about any exceptions the method could possibly generate.

**Create a UML class diagram for Employee class in MS Visio.**

Next, you would implement a client program *EmployeeTest* to test the Employee class. Do the following the EmployeeTest class.

* Implement the main method
* Call the *getEmployeeCount( )* method and print its return value to the output window.
* Create three Employee objects and call toString( ) method on each of the objects to print the output to the output window.
* Display the current count of Employee objects by calling *getEmployeeCount( )* method.
* Demonstrate the usage of the equals( ) method.
* **Make sure to keep make your output verbose and comment your code.**

**Things to turn in:**

1. Type answers to questions in Part – I and II in a word document.
2. Copy and paste the source codes of all the classes and output of the client programs
3. UML Class diagram of Employee class
4. Turn in your assignment on the blackboard by end of the due date.