

CE251 Java Programming Assignment
(Class Fundamental and Multi-Threading)

Due Date: 12/10/2019

- **Write Theory assignment in File pages or in Notebook**
- **Submit Programming assignment in .PDF file only.**

	Theory Assignment
1	Discuss parameter passing by value and by reference in Java with suitable example.
2	Differentiate between protected and default access specifiers.
3	Differentiate between Interface and abstract class. When Interface is preferred over abstract class.
4	State whether each of the following is true or false. If false, explain why. a) If a class contains only private data fields and no setter methods, is the class immutable? b) If all the data fields in a class are private and of primitive types, and the class doesn't contain any setter methods, is the class immutable?
5	What are Reentrant Locks? What are the benefits/disadvantages of Reentrant Locks over synchronization?
6	What is Lock(), UnLock(), ReentrantLock(), TryLock() and How it's different from Synchronized Block in Java?
7	What is an Enum? How do you create an enum from a String value? What is an Enum Ordinal? How do you compare two Enums? Can you use a Switch Statement around an Enum?
8	Explain carefully what null means in Java, and why this special value is necessary.
9	What are the advantages of thread-based multitasking as compared to process-based multitasking?
10	What are the benefits of using a thread pool? How do you create a thread pool with three fixed threads? How do you submit a task to a thread pool? How do you know that all the tasks are finished? [Refer Executors and newFixedThreadPool(int) method creates a fixed number of threads in a pool.]

	Programming Assignment [Submit only in .PDF file]
1	For this problem, you should write a very simple but complete class. The class represents a counter that counts 0, 1, 2, 3, 4, The name of the class should be Counter. It has one private instance variable representing the value of the counter. It has two instance methods: increment() adds one to the counter value, and getValue() returns the current counter value. Write a complete definition for the class, Counter.
2	Write an application that uses objects to model the solar System. Create a class named Planet to encapsulate the name of a planet and the number of its moons. Another class named SolarSystem holds reference to nine planets. Both of these classes provide a method named display() that outputs the encapsulated information. Instantiate the SolarSystem class and invoke its display() method.
3	Write an abstract class named Person and its two subclasses named Student and Faculty. A person has a name, address, phone number, and email address. A student has enrollmentID, courseName. An employee has an office, salary, and designation. Define constructors and methods for input and display for both classes. Write a main program to give demonstration of all.
4	Write a Java class Author with data member firstName and lastName; create getter/setter method for initializing and accessing them. The program also include Book class with title, author and price data member; create getter/setter method for initializing and accessing them. Write a separate class BookDemo with a main() method creates a Book titled "Developing Java Application" with authors Russel Winderand price 79.75. Prints the Book's string representation to standard output (using System.out.println).
5	Write fork join code to find the Fibonacci number by using recursive loop. [Using RecursiveAction]
6	Write fork join code to sum all the numbers from a range. [Using RecursiveTask]
7	Write a code segment that replaces any number of multiple spaces connected together in a String reference s with single spaces. For example, if s contains: "Java is platform independent language.", it should be changed to: "Java is platform independent language."
8	Every method in this problem receives two arrays of int as their input parameters. a. Write combine() which returns a new array whose elements are taken from both input arrays and their orders are preserved starting from the elements from the first input array followed by the ones from the second. b. Write union() which returns a new array whose elements are unique elements taken from both input arrays. The elements of the output array should be sorted increasingly. c. Write intersect() which returns a new array where every elements in the array must be unique and appear in both input arrays. The elements of the output array should be sorted increasingly.

	<p>d. Write <code>subtract()</code> which returns a new array whose elements are unique and appear in the first input array but not in the second one. The elements of the output array should be sorted increasingly.</p> <p>e. Write <code>xor()</code> which returns a new array whose elements are unique and appear in either one of the input arrays but not both. The elements of the output array should be sorted increasingly.</p>
9	<p>Suppose that two arrays of <code>int</code> are said to be equal if they have similar lengths and every elements in the same positions of the two arrays are equal. Write a method called <code>isEqual()</code> which returns <code>true</code> if its two input arrays equal and <code>false</code> otherwise.</p>
10	<p>Create a <code>Circle</code> class that has three private final instance variables named <code>x</code>, <code>y</code>, and <code>radius</code>. The <code>x</code> and <code>y</code> instance variables represent the <code>x</code> and <code>y</code> coordinates of the center of the circle; they are of <code>int</code> data type. The <code>radius</code> instance variable represents the radius of the circle; it is of the <code>double</code> data type. Add a constructor to the <code>Circle</code> class that accepts the values for its instance variables <code>x</code>, <code>y</code>, and <code>radius</code>. Add getters for the three instance variables.</p> <p>Enhance the <code>Circle</code> class by adding four instance methods named <code>centerDistance</code>, <code>distance</code>, <code>overlaps</code>, and <code>touches</code>. All these methods accept a <code>Circle</code> as a parameter. The <code>centerDistance</code> method returns the distance (as a <code>double</code>) between the centers of the circle and another circle passed in as the parameter. The <code>distance</code> method returns the minimum distance (as a <code>double</code>) between the two circles. If two circles overlap, the distance method returns a negative number. The <code>overlaps</code> method returns <code>true</code> if two circles overlaps, <code>false</code> otherwise. The <code>touches</code> method returns <code>true</code> if two circles touches each other, <code>false</code> otherwise. The distance method must use the <code>centerDistance</code> method. The body of the <code>overlaps</code> and <code>touches</code> methods must contain only one statement that uses the distance method.</p> <p>Hint: The distance between two circles is the distance between their centers minus their radii. Two circles overlap if the distance between them is negative. Two circles touch if the distance between them is zero.</p>