CENG 211 Programming Fundamentals
Midterm Exam Questions from Previous Years

Questions:

1. **The Department of Highways is installing a toll collection system. Trucks pulling up to a tollbooth are required to pay a toll of 5 YTL per axle plus 2 YTL per ton of the truck's total weight. A display in the booth shows the toll receipts and the number of truck arrivals since last collection. Write a Java program composed of Truck, TollBooth, TestHelper, and TestTollBoth classes.**

2. **Which integer between 1 and 10000 has the largest number of divisors, and how many divisors does it have? Write a Java program to find the answers and print out the results. It is possible that several integers in this range have the same, maximum number of divisors. Your program has to print out all of them.**

3. **Write a Java program that simulates rolling a pair of dice until the total on the dice comes up to be a given number. The number that you are rolling for should be a parameter to a method. The parameter should be one of the possible totals: 2, 3, ..., 12. The method should throw an IllegalArgumentException if this is not the case.**

4. **Write a Java program that runs some experiments to determine the distribution of numbers returned by Math.random() method, which generates pseudo-random numbers in the range [0,1). Divide this range into N intervals and call the method M times to produce a table with two columns; range and number of occurences, where N and M are inputs from console.**

5. **A meal is either breakfast, lunch, dinner, or snack. Suppose that a person can eat at least 3 meals and at most 5 meals during a day. A lunch is composed of sandwich and dessert, whereas a dinner is composed of soup, salad, meatballs, and dessert. Write a Java program composed of Meal, Breakfast, Lunch, Dinner, Snack, and CalorieCalculator classes, where CalorieCalculator's main method calculates total calorie taken during a day using polymorphism.**

6. **Write Table, Person, and Simulation classes in Java that keep track of persons on a U shape table. The table have a capacity of 2N+1 person. One person sits at the head of the table and there are N seats on each side of the table. There is a rule that persons preferring cheese sit on one side of the table and persons preferring dessert sit on the other side of the table. The person, who does not like either cheese or dessert, sits at the head of the table. The table class's methods should include: (a) a constructor; (b) a method that returns the number of persons sitting on the table or the number of persons with the same taste; (c) a method that adds a new person to the table; (d) a method that removes a certain person (known by a PID) or a group of person with the same taste from the table; (e) a method that prints out the table.**

7. **Suppose that names arraylist contains names, which are of type String. Write a method in Java that removes the names having the given character from the arraylist. Use indexOf(int ch) method of String class, which returns the index (an integer value) within this string of the first occurrence of the specified character or -1 if the character does not occur.**

8. **A truck company has at most 10 trucks. In a day, each truck of this company loads a cargo from a city and carries it to another city using a route. If the cargo is heavier than 1 ton, the company charges 200 TL otherwise it charges 100 TL. The company charges for each 100 km an extra 100 TL. Write necessary classes including Simulation class and methods in Java for the calculation of daily performance (total weight carried, total distance traveled, total amount charged) of the company.**

9. **The naive way of computing a value to the power n performs n−1 multiplications. A much better algorithm can involve repeated squaring:**

   pow(a, 1) = a

   pow(a, n) = pow(a*a, n/2) [n even]

   = a * pow(a*a, (n-1)/2) [n odd]

Even this is not always optimal. For example, it computes a^{15} via the sequence a, a^2, a^3, a^6, a^7, a^{14}, a^{15} in 6 steps while it is possible to use a, a^2, a^4, a^5, a^{10}, a^{15} and manage in 5 steps. Write the necessary Java method(s) that takes an integer n and finds the smallest number of multiplications that could be used to raise a value to the power n.

10. **Write a Java program that simulates tossing a coin repeatedly until the last, say nth, coin's face value is same as the (n-1)th value and also same as (n-2)th value. The simulation class should write each face value to screen. When the last three face values are the same, it should write the number of elements in the coin tossing series. Make sure that your solution has Coin Class and this class contains toss, isEqualTo, toString methods as well as other necessary methods.**