Java Programming
CSPC 23
Assignment 2

1. Which of the following are valid Java identifiers?
   a. `Remainder`
   b. `9_multiple`
   c. `addTwoIntegers`
   d. `PI_VALUE`
   e. `highestof3`
   f. `sum&product`
   g. `diff_&_div`
2. Write a program that displays the word "HI" in large block letters. Make each large letter out of the other character as shown below:

```
I  I  I        I  I  I          H  H  H  H  H  H  H  H  H
I  I  I        I  I  I                   H
I  I  I        I  I  I                   H
I  I  I        I  I  I                   H
I  I  I  I  I  I  I  I  I                H
I  I  I        I  I  I                   H
I  I  I        I  I  I                   H
I  I  I        I  I  I                   H
I  I  I        I  I  I          H  H  H  H  H  H  H  H  H
```

3. Write a program that reads values representing the weight in kilograms, grams, and milligrams and then prints the equivalent weight in milligrams. (For example, 1 kilogram, 50 grams, and 42 milligrams is equivalent to 10,50,042 milligrams.)
4. Write a program that determines the value of the coins in a jar and prints the total in dollars and cents. Read integer values that represent the number of quarters, dimes, nickels, and pennies.
5. Write a line of code that changes all commas(,) in a `String` names sentence into colons(:). Store the modified `String` in a new variable named corrected.
6. Assuming that a Random object has been created called generator, what is the range of the result of each of the following expressions?
   a. `generator.nextInt(20)`
   b. `generator.nextInt(8) + 1`
   c. `generator.nextInt(12) + 2`
   d. `generator.nextInt(35) + 10`
   e. `generator.nextInt(100) – 50`

7. Write code to declare and instantiate an object of the `Random` class (call the object reference variable `rand`). Then write a list of expressions using the `nextInt` method that generates random numbers in the following specified ranges, including the endpoints. Use the version of the `nextInt` method that accepts a single integer parameter.
   a. `0 to 10`
   b. `0 to 400`
   c. `1 to 10`
   d. `1 to 400`
   e. `25 to 50`
   f. `-10 to 15`
8. Write a program that generates a random password that meets certain criteria. The password should start with 7 or 8 or 9, and the next five digits can be any digit from 0 to 9. This should be followed by a dash (-) and then three random uppercase letters. Hint: The integers from 65 to 90 represent the uppercase letters from A to Z. You can cast an integer to a `char` type like this: `(char) 65 is 'A' and (char)66 is 'B'`
9. Write a program that generates a random integer in the range 20 to 40, inclusive, and displays the sine, cosine, and tangent of that number.
10. Write a class called `NumberOfGoals` that represents the total number of goals scored by a football team. The `NumberOfGoals` class should contain a single integer as instance data, representing the number of goals scored. Write a constructor to initialize the number of goals to zero. Write a method called `setGoal` that increments the value by one whenever a goal is scored, and another method called get Goal that returns the total number of goals scored so far. Finally, create a driver class called `GoalTracker` that creates a few `NumberOfGoals` objects and tests their methods.
11. Write a class called Duration that represents a duration of time in hours and minutes. It should contain instance data that represents the start hour, end hour, start minute, and end minute. Define the Duration constructor to accept and initialize all instance data. Include getter and setter methods for all instance data. Include a `toString` method that returns a String in the format [HH:MM,HH:MM]. Write a method called length that returns the length of the duration in minutes. Assume that a 24-hour format is used for all durations of time and no duration will span over multiple days. Create a driver class called `DurationTest`, whose main method instantiates and updates several Duration objects.
12. Write a program that plays the Hi-Lo guessing game with numbers. The program should pick a random number between 1 and 100 (inclusive), then repeatedly prompt the user to guess the number. On each guess, report to the user that he or she is correct or that the guess is high or low. Continue accepting guesses until the user guesses correctly or chooses to quit. Use a sentinel value to determine whether the user wants to quit. Count the number of guesses and report that value when the user guesses correctly. At the end of each game (by quitting or a correct guess), prompt to determine whether the user wants to play again. Continue playing games until the user chooses to stop.

13. Write a program that plays the Rock-Paper-Scissors game against the computer. When played between two people, each person picks one of three options (usually shown by a hand gesture) at the same time, and a winner is determined. In the game, Rock beats Scissors, Scissors beats Paper, and Paper beats Rock. The program should randomly choose one of the three options (without revealing it), then prompt for the user's selection. At that point, the program reveals both choices and prints a statement indicating if the user won, the computer won, or if it was a tie. Continue playing until the user chooses to stop, then print the number of user wins, losses, and ties.
14. Write a program that reads a string from the user, then determines and prints how many of each lowercase vowel (a, e, i, o, and u) appear in the entire string. Have a separate counter for each vowel. Also count and print the number of `non-vowel` characters.
15. Write a method called pad that accepts a string called `str` and an integer called `length`. Return a string with length number of characters by adding an appropriate number of '0' characters to the end of the string. If the string is longer than length, then return the original string unchanged. For example, pad `("aaa",6)` will return "`aaa000`".
16. Write a program that reads an integer value and prints the sum of all even integers between 2 and the input value, inclusive. Print an error message if the input value is less than 2. Prompt accordingly.
17. Write a method called splice that takes a String and two integers named `startingIndex` and deletelength. Remove `deleteLength` characters from the String, starting with index `startingIndex`. Return the resulting string. If `startingIndex` is less than 0, then start removing characters from index 0. If `deleteLength` is 0, then return the original string.
18. Create a class called `SalesPerson` that represents a salesperson in an organization. The `SalesPerson` class should have the name, the phone number, and the assigned district of a salesperson. Each salesperson acquires a certain sale amount every day. Provide a constructor that sets all instance values based on parameter values. Overload the constructor such that each daily sale amount for a week is set to zero. Provide a method called `setDailyAmount` that accepts a day of a week as a number (0 for Sunday, 1 for Monday, and so on) and an amount and sets that amount as the daily sale amount. Provide another method called `getDailyAmount` that accepts a day of a week and returns the amount for that day. Provide a method called total that calculates the total sale amount for the week a method called average that calculates the average daily sale amount for the week. Write a `toString` method that prints all the details of the salesperson. Write a `toString` method that prints all the details of the salesperson. Write a driver class to exercise the `SalesPerson` class.
19. Write a class called `SalesTeam` that represents a team of salespeople. Each salesperson on the team is represented by the `SalesPerson` class as in the previous question. Each team has a name, and the constructor needs only to accept the name of the team. Use and `ArrayList` to store the team members. Provide a method called `addSalesPerson` that accepts a `SalesPerson` object. Provide a method called `weeklyReport` that prints the name and the total sale amount of each team member and the total amount for the entire team. Write a driver class that fully exercise the `SalesTeam` class.