Assignment (Part-3)
Control Statements: Part 1; Assignment, ++ and -- Operators

20. **(Credit Limit Calculator)** Develop a Java application that determines whether any of several department-store customers has exceeded the credit limit on a charge account. For each customer, the following facts are available:

    (a) account number
    (b) balance at the beginning of the month
    (c) total of all items charged by the customer this month
    (d) total of all credits applied to the customer's account this month
    (e) allowed credit limit

    The program should input all these facts as integers, calculate the new balance (= *beginning balance + charges – credits*), display the new balance and determine whether the new balance exceeds the customer's credit limit. For those customers whose credit limit is exceeded, the program should display the message `"Credit limit exceeded"`.

21. **(Sales Commission Calculator)** A large company pays its salespeople on a commission basis. The salespeople receive $200 per week plus 9% of their gross sales for that week. For example, a salesperson who sells $5000 worth of merchandise in a week receives $200 plus 9% of $5000, or a total of $650. You've been supplied with a list of the items sold by each salesperson. The values of these items are as follows:

    ```
    Item  Value
    1     239.99
    2     129.75
    3      99.95
    4     350.89
    ```

    Develop a Java application that inputs one salesperson's items sold for last week and calculates and displays that salesperson's earnings. There's no limit to the number of items that can be sold.

22. **(Salary Calculator)** Develop a Java application that determines the gross pay for each of three employees. The company pays straight time for the first 40 hours worked by each employee and time and a half for all hours worked in excess of 40. You're given a list of the employees, their number of hours worked last week and their hourly rates. Your program should input this information for each employee, then determine and display the employee's gross pay.

23. **(Find the Largest Number)** Write an application that inputs a series of 10 integers and determines and prints the largest integer. Your program should use at least the following three variables:

    (a) `counter`: A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed).
    (b) `number`: The integer most recently input by the user.
    (c) `largest`: The largest number found so far.

24. **(Tabular Output)** Write a Java application that uses looping to print the following table of values:

```
N      10*N        100*N       1000*N
1      10          100         1000
2      20          200         2000
3      30          300         3000
4      40          400         4000
5      50          500         5000
```

25. **(Find the Two Largest Numbers)** Write an application to find the two largest values of the 10 values entered. *[Note: You may input each number only once.]*

26. **(Square of Asterisks)** Write an application that prompts the user to enter the size of the side of a square, then displays a hollow square of that size made of asterisks. Your program should work for squares of all side lengths between 1 and 20.

27. **(Palindromes)** A palindrome is a sequence of characters that reads the same backward as forward. For example, each of the following five-digit integers is a palindrome: 12321, 55555, 45554 and 11611. Write an application that reads in a five-digit integer and determines whether it's a palindrome. If the number is not five digits long, display an error message and allow the user to enter a new value.

28. **(Printing the Decimal Equivalent of a Binary Number)** Write an application that inputs an integer containing only 0s and 1s (i.e., a binary integer) and prints its decimal equivalent. [Hint: Use the remainder and division operators to pick off the binary number's digits one at a time, from right to left. In the decimal number system, the rightmost digit has a positional value of 1 and the next digit to the left a positional value of 10, then 100, then 1000, and so on. The decimal number 234 can be interpreted as 4 * 1 + 3 * 10 + 2 * 100. In the binary number system, the rightmost digit has a positional value of 1, the next digit to the left a positional value of 2, then 4, then 8, and so on. The decimal equivalent of binary 1101 is 1 * 1 + 0 * 2 + 1 * 4 + 1 * 8, or 1 + 0 + 4 + 8 or, 13.]

29. **(Checkerboard Pattern of Asterisks)** Write an application that uses only the output statements

```
System.out.print( "* " );
System.out.print( " " );
System.out.println();
```

to display the checkerboard pattern that follows. A `System.out.println` method call with no arguments causes the program to output a single newline character. *[Hint: Repetition statements are required.]*

```
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
```

30. **(Sides of a Triangle)** Write an application that reads three nonzero values entered by the user and determines and prints whether they could represent the sides of a triangle.

31. **(Sides of a General Right Triangle)** Write an application that reads three nonzero integers and determines and prints whether they could represent the sides of a general right triangle.

32. **(Factorial)** Write an application that reads a nonnegative integer and computes and prints its factorial.

    For example, 5! = 5 · 4 · 3 · 2 · 1, which is 120.