CJ-O-03
Assignment No. 1

1. Accept a integer number and when the program is executed print the binary,octal and hexadecimal equivalent of the given number.

Sample Output:

```
java Test
Enter Number : 20
Given Number :20
Binary equivalent :10100
Octal equivalent :24
Hexadecimal equivalent :14
```

Note:-Refer Java Documentation and look for the appropriate Wrapper class method

2. Write a program to accept 3 digits and print all possible combination of these three digits.
(For example if three digits are 1, 2 and 3 then all possible combinations are 123,132,231,213,321 and 312)

3. Create a BMI (Body Mass Index) calculator that reads the user’s weight in pounds and height in inches (or, if you prefer, the user’s weight in kilograms and height in meters), then calculates and displays the user’s body mass index.
The formula for calculating BMI is

```
          (WeightInKiloGrams)
BMI = ----------------------------
     (HeightInMeters * HeightInMeters);
```

BMI VALUES
Underweight if BMI is less than 18.5
Normal if BMI is in between 18.5 and 24.9
Overweight if BMI is in between 25 and 29.9
Obese if BMI is 30 or greater

4. Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables—a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable.

In addition, provide a method named getInvoiceAmount that

calculates the invoice amount (i.e. multiplies the quantity by the price per item), then returns the amount as a double value.

If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0.

Write a test application named InvoiceTest that demonstrate class Invoice's capabilities.

5.Create a class called Employee that includes three instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Provide a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, do not set its value. Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again.

6. ***Create a class called Date that includes three instance variables—a month (type int), a day (type int) and a year (type int). Provide a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/).***
Write a test application named DateTest that demonstrates class Date's capabilities.

7. (Target-Heart-Rate Calculator) While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your trainers and doctors. According to the American Heart Association (AHA), the formula for calculating your maximum heart rate in beats per minute is 220 minus your age in years. Your target heart rate is a range that's 50–85% of your maximum heart rate. [Note: These formulas are estimates provided by the AHA. Maximum and target heart rates may vary based on thehealth, fitness and gender of the individual. Always consult a physician or qualified health care professional before beginning or modifying an exercise program.]

Create a class called HeartRates.The class attributes should include the person's first name, last name and date of birth (consisting of separate attributes for the month, day and year

of birth). Your class should have a constructor that receives this data as parameters. For each attribute provide set and get methods. The class also should include a method that calculates and returns the person’s age (in years), a method that calculates and returns the person’s maximum heart rate and a method that calculates and returns the person’s target heart rate. Write a Java application that prompts for the person’s information, instantiates an object of class HeartRates and prints the information from that object—including the person’s first name, last name and date of birth—then calculates and prints the person’s age in (years), maximum heart rate and target-heart-rate range.

8. (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".

9. (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	99.99
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.