1. Write a program for calculator –Addition, subtraction, multiplication and division.
2. WAP to swap two numbers using third variable.
3. WAP to swap two numbers without using third variable.
4. Check that given number is even or not.
5. Calculate the factorial of a given number.
6. Check that given number is Prime or not.
7. Display first 10 prime numbers.
8. Reverse the given number.
9. Check that given number is palindrome or not.
10. Check that given number is Armstrong or not.Eg-153=13+53+33=1+225+27=153
11. **WAP for constructor types.**
12. **WAP for method types**
13. **WAP to display information of 3 employees, take the data from user.**
14. **WAP to display the use of access specifiers.**
15. Java Program to Find Out the Number of Objects Created of a Class.
16. [**Java Program to Check Whether a Character is a Vowel, Consonant or Digit**](https://www.sanfoundry.com/java-program-check-given-character-vowel-consonant/)
17. [**Java Program to Check Whether a Given Alphabets are Uppercase or Lowercase or Digits**](https://www.sanfoundry.com/java-program-check-alphabets-uppercase-lowercase-alphabets-digits/)
18. Display the sum of digits of a number.
19. Calculate the power of a number without power function and with power function.
20. Java program to check leap year.
21. Write a Java program to create and display unique three-digit number using 1, 2, 3, 4. Also count how many three-digit numbers are there. 123
22. Write a Java program to print the ASCII value of a given character.
23. Finding square root of a number. Program to print a given number's square root without employing the math.sqrt() function.
24. **WAP to accept the array elements from user and display the elements.**
25. **Write a Java Program to find the highest number and lowest number in an array.**
26. **Program to add any two given matrices and print the result.**
27. **Program to multiply any two given matrices and print the result.**
28. **Transposing Matrix. Program to print the transpose of a given matrix. A transpose of a matrix has all its rows and columns interchanged.**
29. **Write a Java Program to find the second-highest number in an array.**
30. **Calculate and return the sum of all the even numbers present in the numbers array passed to the method calculateSumOfEvenNumbers. Implement the logic inside calculateSumOfE venNumbers() method.Test the functionalities using the main() method of the Tester class.**
31. **Program to identify and remove all repeated elements from an array.**
32. **Write a Java program that demonstrate the the use of method overloading, method overriding and method hiding.**
33. **Write a Java program to find the common elements between two arrays of string (Hint:equals())**
34. **Write a Java program to find the common elements between two arrays of integers**
35. **Check the equality of two arrays.**
36. **Write a Java program to create an array of its anti-diagonals from a given square matrix.**
37. **Write a Java program to segregate all 0s on left side and all 1s on right side of a given array of 0s and 1s.**
38. [Java Program to Count Number of Duplicate Words in String](https://www.javaguides.net/2018/08/java-program-to-count-number-of-duplicate-words-in-string.html)
39. Java Program to Reverse a String.
40. How to Check if the String Contains e in umbrella
41. Check word orange in “This is Orange juice”.
42. Remove all whitespace. VIT Pune-VITPune
43. [Java Program to Check if Input String is Palindrome](https://www.javaguides.net/2018/08/java-program-to-check-if-input-string-is-palindrome.html)
44. Write a Java program to count the number of words in a string?
45. Write a Java program to check whether two strings are anagram or not?
46. Write a java program that accept your full name as input and display initial of first and middle name and complete last name . Varsha Rahul Dange-V. R. Dange
47. WAP to demonstrate the use of throws and throw in exception handling.
48. Create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.
49. Write a Java program which creates only one object. If user attempts to create second object, he should not be able to create it. Display appropriate meassage (Using Exception Handling).
50. A Company manufactures Vehicles, which could be a Helicopter, a Car, or a Train depending on the customer’s demand. Each Vehicle instance has a method called move, which prints on the console the nature of movement of the vehicle. For example,

the Helicopter Flies in Air,

the Car Drives on Road and

the Train Runs on Track.

Write a program that accepts input from the user on the kind of vehicle the user wants to order, and the system should print out nature of movement. Implement all Java coding best practices to implement this program.

1. Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.
2. We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for eac of the two classes and print the percentage of marks for both the students.
3. An abstract class has a construtor which prints "This is constructor of abstract class", an abstract method named 'a\_method' and a non-abstract method which prints "This is a normal method of abstract class". A class 'SubClass' inherits the abstract class and has a method named 'a\_method' which prints "This is abstract method". Now create an object of 'SubClass' and call the abstract method and the non-abstract method. (Analyse the result)

54.

**Problem Description**: Peter wants to start a payment service portal for making payments for the credit card bill and online shopping. He will need a base class (**RRPaymentServices**) to store balance and customer ID information. He will use two classes- **CreditCardPayment** and **ShoppingPayment** for paying Credit card bills and shopping bills respectively. The bill payment and id generation is different for both payment modes.

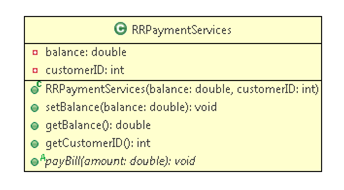
**Class Description:**

Create a Java Project with the name **AbstractClassesAndMethods**

**RRPaymentServices:**

This is the base class for CreditCardPayment and ShoppingPayment classes.

* It will store the amount to be paid by the user in the instance variable balance of the class.

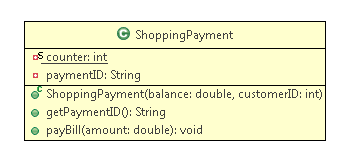


**Method Description:**

* payBill(double amount): This is an abstract method that has to be implemented by the child classes of RRPaymentServices.

**ShoppingPayment:**

This class is a child class of RRPaymentServices. It has a static variable **counter,** to set the **paymentID.** The class diagram is shown below:

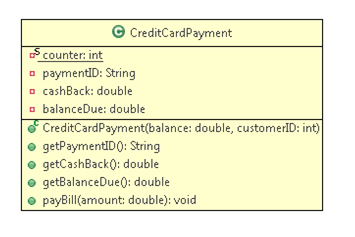


**Method Description:**

* payBill(double amount): For a shopping bill payment, the payment id should start with 'S' followed by a four digit integer number. If the user enters an amount not equal to the balance which is due, an appropriate error message should be displayed and the id should get generated only for valid payments.

**CreditCardPayment:**

This class is another child class of RRPaymentServices. It has a static variable **counter,** to generate the **paymentID.** The class diagram is shown below:

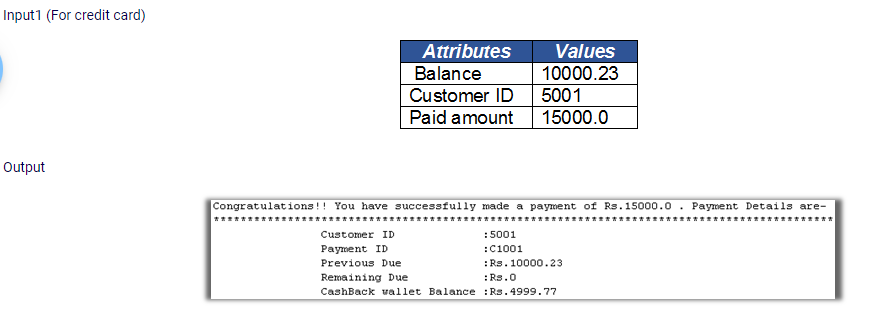


**Method Description:**

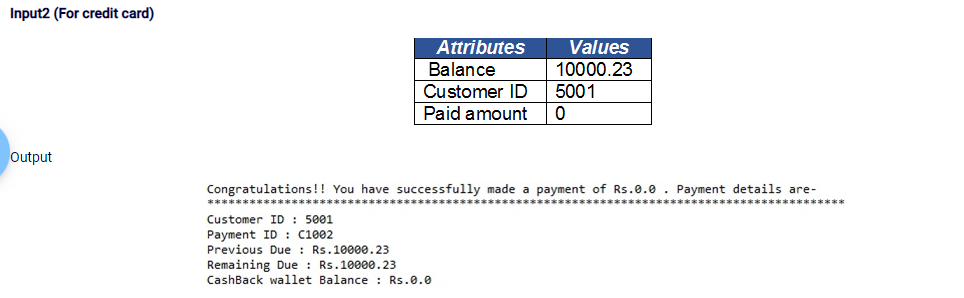
* payBill(double amount): For credit card bill payment, if a user enters an amount more than the amount to be paid, the excess amount should be stored as cashBack. The payBill method is used to pay the bill and generate the transaction id. The id should start with 'C' followed by a four-digit integer number. In case the user enters an amount less than the amount to be paid, then the remaining amount should be stored in the instance variable balanceDue of class CreditCardPayment.
* Use a Tester class to test your code and display the payment information.
* The balanceDue after the payment should be communicated to the user.

Some outputs with different inputs are given below for your reference.

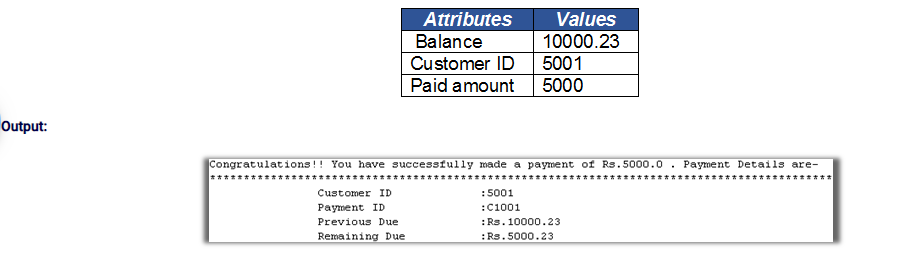
**Input1 (For credit card)**



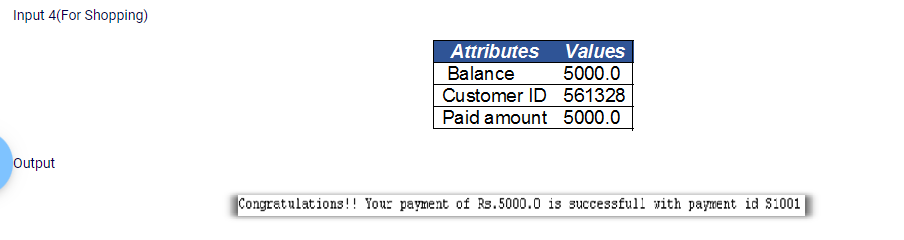
**Input2 (For credit card)**

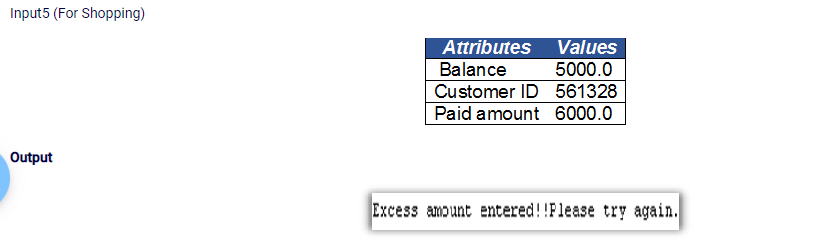


**Input3 (For Shopping)**

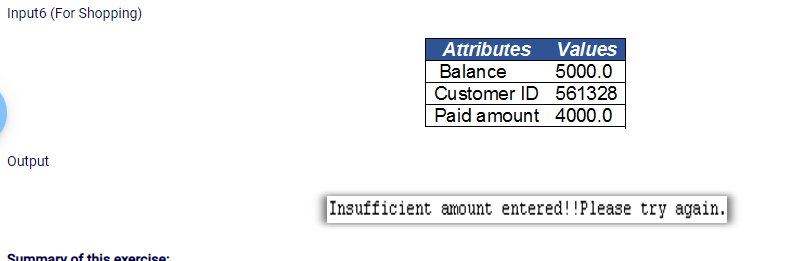
****

**Input 4(For Shopping)**



**Input5 (For Shopping) **

**Input6 (For Shopping)**



1. **Write a program to traverse (or iterate) ArrayList?**
2. **Write a program to convert List to Array.**
3. **Given an element write a program to check if element(value) exists in ArrayList?**
4. Create arrayList, add the integer elements in arrayList using asList().Remove the duplicate values and return a arrayList containing unique values. Implement the logic inside removeDuplicates() method. Test the functionalities using the main () method of the Tester class. Sample Input and Output---------10, 20, 10, 15,40,15,40 --- 10,20,15,40
5. Given two lists, concatenate the second list in reverse order to the end of the first list and return the concatenated list. Implement the logic inside concatenateLists() method. listOne = Hello 102 200.8 25 listTwo = 150 40.8 welcome A output: Hello 102 200.8 25 A welcome 40.8 150
6. Write a Java Program to iterate ArrayList using for-loop, iterator, and advance for-loop. Insert 3 Array List. Input : 20 30 40 Output: Iterator Loop: 20 30 40 Advanced For Loop: 20 30 40 for Loop: 20 30 40.
7. replace all the occurrences of an element in a list.
8. Define an ArrayList of String and add any four names to it. Using ListIterator, traverse through the ArrayList first in forward and then in the reverse direction, and print the names to the console.

**Exercise Description:**

Define a LinkedList of String and add any four names to it.

Using 'for loop', traverse through the array and print the names to the console.

Remove the first and the last element and print the list using 'advanced for loop'.

Add new names at the first and last position and print the list using an iterator.

1. Create a Java program that adds Student objects into a HashSet. The Student class has name and rollNumber as its attributes and a toString() method. If we add two Student objects having the same rollNumber to the HashSet, it should be considered as duplicate and should not get added.//sample code

public class Student{

private String name;

private int rollNumber;

public Student(String name, int rollNumber) {

this.name = name;

this.rollNumber = rollNumber;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getRollNumber() {

return rollNumber;

}

public void setRollNumber(int rollNumber) {

this.rollNumber = rollNumber;

}

//add toString() method

// add appropriate method for avoiding duplicate Student

}

Modify the Student class created in 'Set-Collections - Exercise' to have an additional attribute totalMarks (not greater than 100). Calculate the grade for a student based on the total marks, as per the below mentioned conditions. Store it in a TreeMap containing roll number as the key and grade as the value.

| **Total Marks** | **Grade** |
| --- | --- |
| >=60 | A |
| <60 and >40 | B |
| <=40 | C |

Then, print the grade and roll number of each student from the Map.

1) Write a Java program to divide two numbers and print on the screen.

2) Write a Java program to print the result of the following operations

Test Data:

a. -7 + 8 \* 6

b. (57+9) % 9

c. 20 + -2\*5 / 8

d. 5 + 16 / 3 \* 2 - 9 % 2

3) Write a Java program to print the sum (addition), multiply, subtract, divide and remainder of two numbers

Test Data:

Input first number: 125

Input second number: 24

Expected Output :

125 + 24 = 149

125 - 24 = 101

125 x 24 = 3000

125 / 24 = 5

125 mod 24 = 5

4) Write a Java program to print the area and perimeter of a circle.

Test Data:

Radius = 7.5

Expected Output

Perimeter is = 47.12 (output display upto 2 places)

Area is = 176.71 (output display upto 2 places)

5) Write a Java program that takes three numbers as input to calculate and print the average of the numbers.

6) Write a Java program to compare two numbers.

Input first integer: 45

Input second integer: 69

Expected Output

45 != 69

45 < 69

45 <= 69

7) Write a Java program and compute the sum of the digits of an integer.

Input Data:

Input an integer: 63

Expected Output

The sum of the digits is: 9

1. **import** java.util.Scanner;
2. **public** **class** Digit\_Sum
3. {
4. **public** **static** **void** main(String args[])
5. {
6. **int** m, n, sum = 0;
7. Scanner s = **new** Scanner(System.in);
8. System.out.print("Enter the number:");
9. m = s.nextInt();
10. **while**(m > 0)
11. {
12. n = m % 10;
13. sum = sum + n;
14. m = m / 10;
15. }
16. System.out.println("Sum of Digits:"+sum);
17. }

8) Write a Java program to create and display unique three-digit number using 1, 2, 3, 4. Also count how many three-digit numbers are there. 123

124

...

431

432

Total number of the three-digit-number is 24

9) Write a Java program to print the ascii value of a given character.

Expected Output

The ASCII value of Z is :90

10) Write a program to check that entered password is correct or not.

11) Write a Java program to print numbers between 1 to 100 which are divisible by 3, 5 and by both.

Sample Output:

Divided by 3:

3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57

, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99,

Divided by 5:

5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90,

95,

Divided by 3 & 5:

15, 30, 45, 60, 75, 90,

12)Write a Java program to convert a string to an integer in Java.

Sample Output:

Input a number(string): 25

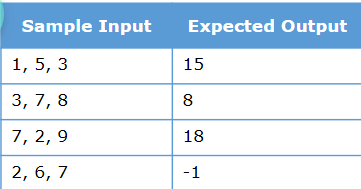
The integer value is: 25

13) Implement a program to display the sum of two given numbers if the numbers are same. If the numbers are not same, display the double of the sum.

14) Implement a program to calculate the product of three positive integer values. However, if one of the integers is 7, consider only the values to the right of 7 for calculation. If 7 is the last integer, then display -1.

Note: Only one of the three values can be 7

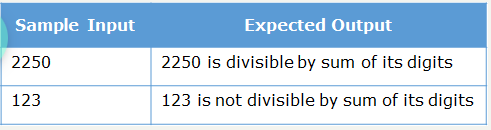
**Sample Input and Output**



15)Implement a program to find out whether a number is divisible by the sum of its digits.

Display appropriate messages.

**Sample Input and Output**



16) Implement a program to check whether a given number is an Armstrong number.

An Armstrong number is an n-digit number that is equal to the sum of the nth powers of its individual digits.

E.g.: 371 is an Armstrong number as 33 + 73 + 13=371

1634 is an Armstrong number as 14 + 64 + 34+ 44=1634

**Hint**

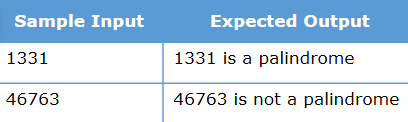
Use Math.pow(double a, double b) method to calculate the power of a number

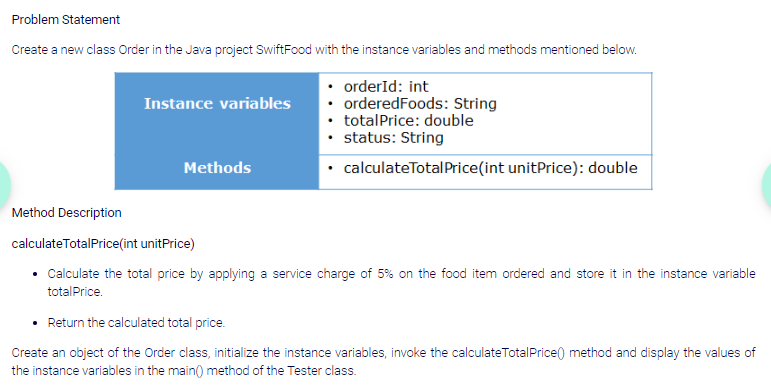
17) Implement a program to check whether a given number is a palindrome.

Palindrome is a sequence that reads the same backwards as forwards.

E.g.: 121, 1331, 2332, 78900987, 123456654321, etc.

**Sample Input and Output**



18) 



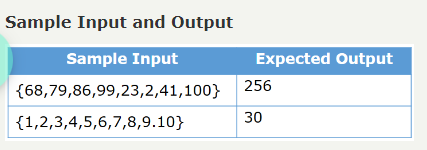
19) the food item ordered gets displayed. If the ordered food is anything other than Burger, Pizza and Sandwich, then a message, 'Invalid selection' is displayed from the default case.

20) write the code to get a 5% discount for Regular customers and 10% for Premium customers. Display related message according customerType

21) Calculate and return the sum of all the even numbers present in the numbers array passed to the method calculateSumOfEvenNumbers. Implement the logic inside

calculateSumOfE venNumbers() method.

Test the functionalities using the main() method of the Tester class.



class Tester {

public static int calculateSumOfEvenNumbers(int[] numbers){

int total=0;

for(int i=0;i<numbers.length;i++)

{

if(numbers[i]%2==0)

{

total+=numbers[i];

}

}

return total;

}

public static void main(String[] args) {

int[] numbers = {68,79,86,99,23,2,41,100};

System.out.println("Sum of even numbers: " +calculateSumOfEvenNumbers(numbers));

}

}