JAVA PROGRAMS

Number Based Logical Programs

Java program to find perfect numbers

Perfect Number: a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself.

```
package demo;
import java.util.Scanner;
public class PerfectNumber
{
 public static void main(String[] args)
 {
  Scanner sc = new Scanner(System.in);
  int n, i = 1, sum = 0;
  System.out.print("Enter a number: ");
  n = sc.nextInt();
  while (i < n) {
   if (n % i == 0) {
    sum = sum + i;
   }
   i++;
  }
  if (sum == n) {
   System.out.print(i + " is a perfect number");
  } else {
   System.out.print(i + " is not a perfect number");
  }
 }
}
```

Output:

```
Enter a number: 6
6 is a perfect number
BUILD SUCCESSFUL (total time: 36 seconds)
```

Java program to print perfect numbers from 1 to 1000

Perfect Number: a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself.

```
package demo;
public class a
{
```

```
 public static void main(String[] args)
 {
  int i, sum;
  System.out.print("Perfect numbers are: ");
  for (int n = 1; n <= 1000; n++)
  {
   i = 1;
   sum = 0;
   while (i < n) {
    if (n % i == 0) {
     sum = sum + i;
    }
    i++;
   }
   if (sum == n) {
    System.out.print(n + " ");
   }
  }
 }
}
```

Output:

```
Perfect numbers are: 6 28 496
BUILD SUCCESSFUL (total time: 2 seconds)
```

Java program to split number into digits.

```
package demo;
import java.util.Scanner;
public class SplitNumber
{
 public static void main(String[] args)
 {
  int num, temp, factor = 1;
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter a number: ");
  num = sc.nextInt();
  temp = num;
  while (temp != 0) {
   temp = temp / 10;
   factor = factor * 10;
  }
  System.out.print("Each digits of given number are: ");
  while (factor > 1) {
   factor = factor / 10;
   System.out.print((num / factor) + " ");
```

```
   num = num % factor;
  }
 }
}
```

Output:
Enter a number: 4732
Each digits of given number are: 4 7 3 2
BUILD SUCCESSFUL (total time: 21 seconds)

Java program to swap two array.

```
package demo;
import java.util.Scanner;
public class SwappingArray
{
 public static void main(String[] args)
 {
  int[] a = new int[10], b = new int[10], c = new int[10];
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter First array: ");
  for (int i = 0; i < 10; i++) {
   a[i] = sc.nextInt();
  }
  System.out.println("Enter Second array: ");
  for (int i = 0; i < 10; i++) {
   b[i] = sc.nextInt();
  }
  System.out.println("Arrays before swapping");
  System.out.println("First array: ");
  for (int i = 0; i < 10; i++) {
   System.out.print(a[i]);
  }
  System.out.println("Second array: ");
  for (int i = 0; i < 10; i++) {
   System.out.print(b[i]);
  }
  for (int i = 0; i < 10; i++) {          //write any swapping technique
   c[i] = a[i];
   a[i] = b[i];
   b[i] = c[i];
  }
  System.out.println("Arrays after swapping");
  System.out.println("First array: ");
  for (int i = 0; i < 10; i++) {
   System.out.print(a[i]);
  }
```

```
  System.out.println("Second array: ");
  for (int i = 0; i < 10; i++) {
   System.out.print(b[i]);
  }
 }
}
```

Output:

```
Enter First array:
1
2
3
4
5
6
7
8
9
10

Enter Second array:
11
22
3
44
55
66
77
55
22
33

Arrays before swapping

First array: 12345678910
Second array: 1122344556677552233

Arrays after swapping
First array: 1122344556677552233
Second array:  12345678910

BUILD SUCCESSFUL (total time: 27 seconds)
```

Java program to find NCR factor of given number.

Calculate the combinations for C(n,r) = n! / (r!(n - r)!). For 0 <= r <= n.

```
package demo;
```

```
import java.util.Scanner;
public class Ncr
{
 public static void main(String[] args)
 {
  Scanner sc = new Scanner(System.in);
  int n, r, ncr;
  System.out.print("Enter any two numbers-");
  n = sc.nextInt();
  r = sc.nextInt();
  ncr = fact(n) / (fact(r) * fact(n - r));
  System.out.print("The NCR factor of " + n + " and " + r + " is " + ncr);
 }

 public static int fact(int n)
 {
  int i = 1;
  while (n != 0) {
   i = i * n;
   n--;
  }
  return i;
 }
}
```

Output:
Enter any two numbers- 5
3
The NCR factor of 5 and 3 is 10
BUILD SUCCESSFUL (total time: 5 seconds)

Java program to find generic root.

Generic Root: It sum of digits of a number unit we don't get a single digit.
Example:
Generic root of 4563: 4+5+6+3 = 18 since 18 is two digit numbers so 1 + 8 = 9
So, generic root of 4563 = 9

```
package demo;
import java.util.Scanner;
public class GenericRoot
{
 public static void main(String[] args)
 {
  Scanner sc = new Scanner(System.in);
  long num, sum = 0, r;
  System.out.print("Enter a number:-");
```

```
  num = sc.nextLong();
  while (num > 10)
  {
   sum = 0;
   while (num != 0)
   {
    r = num % 10;
    num = num / 10;
    sum += r;
   }
   if (sum > 10)
   {
    num = sum;
   } else
   {
    break;
   }
  }
  System.out.println("Sum of the digits in single digit is: " + sum);
 }
}
```

Output:

```
Enter a number:-5555
Sum of the digits in single digit is: 2
BUILD SUCCESSFUL (total time: 6 seconds)
```

Java program to add two number without using addition operator.

```
package demo;
import java.util.Scanner;
public class Addition
{
 public static void main(String[] args)
 {
  int a, b;
  int sum;
  Scanner sc = new Scanner(System.in);
  System.out.print("Enter any two integers: ");
  a = sc.nextInt();
  b = sc.nextInt();
  sum = a - ~b - 1;
  System.out.print("Sum of two integers: " + sum);
 }
}
```

Output:

```
Enter any two integers: 20
50
Sum of two integers: 70
BUILD SUCCESSFUL (total time: 10 seconds)
```

Algorithm:

In c ~ is 1's complement operator. This is equivalent to:

~a = -b + 1

So, a - ~b -1

= a-(-b + 1) + 1

= a + b - 1 + 1

= a + b

Java program to sum digits of a number.

```
package demo;
import java.util.Scanner;
public class SumOfDigits
{
 public static void main(String[] args)
 {
  Scanner sc = new Scanner(System.in);
  int num, sum = 0, r;
  System.out.println("Enter a number: ");
  num = sc.nextInt();
  while (num != 0) {
   r = num % 10;
   num = num / 10;
   sum = sum + r;
  }
  System.out.print("Sum of digits of number:  " + sum);
 }
}
```

Output:

```
Enter a number: 32165
Sum of digits of number:  17
BUILD SUCCESSFUL (total time: 8 seconds)
```

Java program to reverse a number.

```
package demo;
import java.util.Scanner;
```

```
public class ReverseNumber
{
 public static void main(String[] args)
 {
  Scanner sc = new Scanner(System.in);
  int num, r, reverse = 0;
  System.out.print("Enter any number: ");
  num = sc.nextInt();
  while (num != 0)
  {
   r = num % 10;
   reverse = reverse * 10 + r;
   num = num / 10;
  }
  System.out.println("Reversed of number: " + reverse);
 }
}
```

Output:

```
Enter any number: 123789
Reversed of number: 987321
BUILD SUCCESSFUL (total time: 4 seconds)
```

String Based Programs

Java program to find the frequency of one string in another string.

```
import java.util.Scanner;
public class WordFrequencyCounter {
    public static void main(String args[]) {
        Scanner scan = new Scanner(System.in);
        System.out.println("enter main string");
        String s = scan.nextLine();
        System.out.println("enter string to be searched");
        String f = scan.nextLine();
        s = s + " ";
        int l = s.length();
        char a;
        int c = 0;
        String s1 = "";
        for (int i = 0; i < l; i++) {
            a = s.charAt(i);
            if (a != ' ') {
```

```
            s1 = s1 + a;
         } else {
            if (s1.equalsIgnoreCase(f) == true) {
               c++;
            }
            s1 = "";
         }
      }
      System.out.println("Frequency of the word " + f + " is " + c);
   }
}
```

Output:

```
Enter main string
Umesh Kumar Kushwaha Kumar
enter string to be searched
Kumar
Frequency of the word Kumar is 2
```

Java program to remove vowels from a string.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class RemoveVowels
{
 public static void main(String args[]) throws IOException {
   BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
   String str1, str2 = "";
   char ch, ch1;
   int i, len;
   System.out.println("Enter the Sentence:");
   str1 = br.readLine();
   len = str1.length();
   for (i = 0; i < len; i++) {
    ch = str1.charAt(i);
    ch1 = Character.toLowerCase(ch);
    switch (ch1) {
     case 'a':
     case 'e':
     case 'i':
     case 'o':
     case 'u':
      break;
     default:
      str2 = str2 + ch;
   }                        //switch ends
```

```
   }                        //for ends
   System.out.println("Modified string without vowels:" + str2);
  }                         //method ends
}
```

Output:

```
Enter the Sentence:
umesh kumar kushwaha
Modified string without vowels:msh kmr kshwh
```

Java program to remove common characters from two strings.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class RemoveCommonDuplicateChar
{
 public static void main(String args[]) throws IOException {
  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
  String s1, s2, s3 = "", s4 = "";
  int a[], b[], x = 0, y = 0;
  int len1, len2;
  char ch;
  int i, j;
  System.out.println("Enter the first sentence:");
  s1 = br.readLine().trim();
  System.out.println("Enter the second sentence:");
  s2 = br.readLine().trim();
  len1 = s1.length();
  len2 = s2.length();
  a = new int[len1];
  b = new int[len2];
  for (i = 0; i < len1; i++) {
   ch = s1.charAt(i);
   for (j = 0; j < len2; j++) {
    if (ch == s2.charAt(j)) {
     break;
    }
   }
   if (j != len2) {
    a[x++] = i;
    b[y++] = j;
   }
  }
  for (i = 0; i < len1; i++) {
   for (j = 0; j < x; j++) {
    if (i == a[j]) {
```

```
     break;
    }
   }
   if (j == x) {
    s3 = s3 + s1.charAt(i);
   }
  }
  for (i = 0; i < len2; i++) {
   for (j = 0; j < y; j++) {
    if (i == b[j]) {
     break;
    }
   }
   if (j == y) {
    s4 = s4 + s2.charAt(i);
   }
  }
  System.out.println("Original string1=" + s1 + " Modified string1=" + s3);
  System.out.println("Original string2=" + s2 + " Modified string2=" + s4);
 }
}
```

Output:

```
Enter the first sentence:
Umesh Kumar Kushwaha
Enter the second sentence:
Rahul Kumar Singh
Original string1=Umesh Kumar Kushwaha Modified string1=Uessw
Original string2=Rahul Kumar Singh Modified string2=Rlua Singh
```

Java program to finding Shortest and Longest Words in a String.

```
import java.io.BufferedReader;
class FindMinMaxString {
 public static void main(String args[]) {
  findMethod("My name is Umesh Kushwaha");
 }
 static public void findMethod(String s) {
  String str = s + " ";
  char ch = ' ';
  int len = str.length(), l = 0;
  int min = len, max = 0;
  String shortest_word = "", longest_word = "", word = "";
  for (int i = 0; i < len; i++) {
   ch = str.charAt(i);
   if (ch != ' ') {
    word += ch;
```

```
   }                                  //if ends
   else {
    l = word.length();
    if (l < min) {
     min = l;
     shortest_word = word;
    }                                   //if ends
    if (l > max) {
     max = l;
     longest_word = word;
    }
    word = "";
   }
  }
  System.out.println("Shortest word = " + shortest_word + " with length " + min);
  System.out.println("Longest word = " + longest_word + " with length " + max);
 }
}
```

Output:
Shortest word = My with length 2
Longest word = Kushwaha with length 8

Java program to reverse a String without using direct method.

```
import java.io.BufferedReader;
public class ReverseOfString {
 public static String reverseString(String s)
 {
  int l = s.length();
  String backward = "";
  for (int i = l - 1; i >= 0; i--)
 {
   backward = backward + s.charAt(i);
  }
  return backward;
 }
 public static void main(String args[]) {
  String backwards = reverseString("Umesh Kushwaha");
  System.out.println("Reverse String is " + backwards);
 }
}
```

Output:
Reverse String is ahawhsuK hsemU

Java program to input name, middle name and surname of a person and print only the initials.

Example.

```
Enter your name
Umesh Kumar Kushwaha
Initials are   U.K.K.
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
class InitialName {
 public static void main(String args[]) throws IOException {
  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
  System.out.println("Enter your name");
  String s = br.readLine(), s1 = "";
  s = " " + s;
  int len = s.length();
  char a;
  for (int i = 0; i < len; i++) {
   a = s.charAt(i);
   if (a == ' ') {
    s1 = s1 + s.charAt(i + 1) + ".";
   }
  }
  System.out.println("Initials are    " + s1);
 }
}
```

Output:

```
Enter your name
Umesh Kumar Kushwaha
Initials are   U.K.K.
```

Java program to remove duplicates characters from given String.

```
class RemoveDuplicate {
 public static void main(String args[]) {
  remove("123aa bbbcc  c2589  99oppq rtyyy");
 }
 static void remove(String s) {
  int l = s.length();
  int c;
  String org = s, s1 = "";
  for (int i = 0; i < (l - 1); i++) {
   s1 = s.substring(0, i + 1);
```

```
   c = 0;
   for (int j = i + 1; j < l; j++) {
    if (s.charAt(i) == s.charAt(j)) {
     c++;
     continue;
    } else
     s1 = s1 + s.charAt(j);
   }
   s = s1;
   s1 = "";
   if (c > 0)
    l -= c;
  }
  System.out.println("Original String:" + org);
  System.out.println("String after removing duplicates:   " + s);
 }
}
```

Output:

Original String:123aa bbbcc c2589 99oppq rtyyy

String after removing duplicates: 123a bc589opqrty

Java program to count the occurrence of any character in given String.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.logging.Level;
import java.util.logging.Logger;

class CountFrequencyOfWord {
 public static void main(String args[]) {
  try {
   input();
  } catch (IOException ex) {
   Logger.getLogger(CountFrequencyOfWord .class.getName()).log(Level.SEVERE, null, ex);
  }
 }
 static void input() throws IOException {
  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
  System.out.println("Enter a sentence");
  String s = br.readLine();
  System.out.println("Enter the letter ");
  char a = (char) br.read();
```

```
  int len = s.length();
  int c = 0;
  char x;
  for (int i = 0; i < len; i++) {
   x = Character.toLowerCase(s.charAt(i));
   if (x == a)
    c = c + 1;
  }
  System.out.println("Frequency of the letter  " + a + " in the sentence " + s + " is " + c);
 }
}
```

Output:

```
Enter a sentence
Hi my name is umesh
Enter the letter
m
Frequency of the letter  m in the sentence "Hi my name is umesh" is 3
```

Java program to find palindromic substring in a string.

```
public class PalidrumSubString
{
 public static void main(String args[])
 {
  String s = "my name rar kumar poop";
  String p[] = s.split(" ");
  String s2 = "";
  for (int j = 0; j < p.length; j++)
  {
   for (int i = p[j].length() - 1; i >= 0; i--)
   {
    char z = p[j].charAt(i);
    s2 = s2 + z;
   }

   if (p[j].equals(s2))
   {
    System.out.println(s2);
   }
   else
   {
    s2 = "";
   }
  }
 }
}
```

Output:

rar
poop

Java program to accept a string and print each character of the string along with its ASCII code.

```
class CalculateAscii {
 public static void main(String args[]) {
  word("Umesh Kushwaha");
 }
 static void word(String s) {
  int len = s.length();
  int a;
  char a1;
  for (int i = 0; i < len; i++) {
   a1 = s.charAt(i);
   a = a1;
   System.out.println(a1 + "->" + a);
  }
 }
}
```

Output:

U->85
m->109
e->101
s->115
h->104
 ->32
K->75
u->117
s->115
h->104
w->119
a->97
h->104
a->97

Common Logical Programs

Java program to find whether given no. is Armstrong or not.

```
Example :
Input - 153
Output - 1^3 + 5^3 + 3^3 = 153, so it is Armstrong no. */

class Armstrong {
 public static void main(String args[]) {
  int num = Integer.parseInt(args[0]);
  int n = num;                              //use to check at last time
  int check = 0, remainder;
  while (num > 0) {
   remainder = num % 10;
   check = check + (int) Math.pow(remainder, 3);
   num = num / 10;
  }
  if (check == n)
   System.out.println(n + " is an Armstrong Number");
  else
   System.out.println(n + " is not a Armstrong Number");
 }
}
```

Output

```
153
153 is an Armstrong Number
```

Java program to Generate prime numbers between 1 & given number

```
/*
Prime Numbers Java Example
This Prime Numbers Java example shows how to generate prime numbers
between 1 and given number using for loop.
*/
public class GeneratePrimeNumbersExample {
 public static void main(String[] args) {
  int limit = 100;                    //define limit
  System.out.println("Prime numbers between 1 and " + limit);
  for (int i = 1; i < 100; i++) {     //loop through the numbers one by one
   boolean isPrime = true;
   for (int j = 2; j < i; j++) {      //check to see if the number is prime
    if (i % j == 0) {
```

```
     isPrime = false;
     break;
    }
   }                                  // print the number
   if (isPrime)
    System.out.print(i + " ");
  }
 }
}

/*
Output of Prime Numbers example would be
Prime numbers between 1 and 100
1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
*/
```

Java program to calculate Factorial of a number using recursion

```
/*
This program shows how to calculate
Factorial of a number using recursion function.
*/
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class JavaFactorialUsingRecursion {
 public static void main(String args[]) throws NumberFormatException,
  IOException {
   System.out.println("Enter the number: ");    //get input from the user
   BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
   int a = Integer.parseInt(br.readLine());
   int result = fact(a);           //call the recursive function to generate factorial
   System.out.println("Factorial of the number is: " + result);
  }
 static int fact(int b) {
  if (b <= 1)                      //if the number is 1 then return 1
   return 1;
  else                             //else call the same function with the value - 1
   return b * fact(b - 1);
 }
}


/*
Output of this Java example would be
Enter the number:
5
Factorial of the number is: 120
*/
```

Java program to check Palindrome Number.

```
/*
This program shows how to check for in the given list of numbers
whether each number is palindrome or not
*/
public class JavaPalindromeNumberExample {
 public static void main(String[] args) {
  int numbers[] = new int[] {121,13,34,11,22,54};
  for (int i = 0; i < numbers.length; i++) {
   int number = numbers[i];
   int reversedNumber = 0;
   int temp = 0;
   /*
    * If the number is equal to it's reversed number, then
    * the given number is a palindrome number.
    *
    * For ex,121 is a palindrome number while 12 is not.
    */
   //reverse the number
   while (number > 0) {
    temp = number % 10;
    number = number / 10;
    reversedNumber = reversedNumber * 10 + temp;
   }
   if (numbers[i] == reversedNumber)
    System.out.println(numbers[i] + " is a palindrome");
   else
    System.out.println(numbers[i] + " not a palindrome ");
  }
 }
}

/*
Output of Java Palindrome Number Example would be
121 is a palindrome number
13 is not a palindrome number
34 is not a palindrome number
11 is a palindrome number
22 is a palindrome number
54 is not a palindrome number
*/
```

Java program to calculate the Simple Interest and Input by the user.

```
import java.util.*;
class SimpleInterest {
```

```
 int p, t;
 float si, r;
 public si() {
  r = 0;
  p = 0;
 }
 public void getdata() {
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter principle : ");
  p = sc.nextInt();
  System.out.println("Enter rate : ");
  r = sc.nextFloat();
  System.out.println("Enter time period : ");
  t = sc.nextInt();
 }
 public void cal() {
  si = (p * r * t) / 100;
 }
 public void display() {
  System.out.println("Principle : Rs" + p);
  System.out.println("Rate : " + r);
  System.out.println("Time period : " + t);
  System.out.println("Simple Interest : Rs" + si);
 }
 public static void main(String args[]) {
  si s = new si();
  s.getdata();
  s.cal();
  s.display();
 }
}
```

Java program to design a class account using the inheritance and static that show all function of bank and generate account number dynamically.

```
import java.util.*;
class Bank {
 static int acc_no = 10001;
 float amt;
 public void display() {
  System.out.println("Account no :" + acc_no);
  System.out.println("Current Amount :" + amt);
 }
 public bank() {
  amt = 1000;
  System.out.println("Ur account no is " + acc_no);
```

```
 acc_no++;
}
public void getamt() {
 System.out.println("Current balance :" + amt);
}
public void withdraw(float x) {
 if (amt == 1000 || amt <= x) {
  System.out.println("Sorry u can't withdraw");
 } else {
  amt = amt - x;
  System.out.println("amount withdrawn :" + x);
  System.out.println("After withdrawl");
  getamt();
 }
}
public void deposit(float x) {
 if (x == 0.0)
  System.out.println("OOPS 0 can't be deposited");
 else {
  amt += x;
  System.out.println("After deposition");
  getamt();
 }
}
public static void main(String args[]) {
 Scanner sc = new Scanner(System.in);
 bank b1 = new bank();
 b1.deposit(0);
 b1.withdraw(120.5 f);
 b1.display();
 System.out.println("\n");
 bank b2 = new bank();
 b2.deposit(1000.0 f);
 b2.withdraw(150.5 f);
}
}
```

Java program to find out the HCF and LCF.

```
import java.util.*;
class Hcf {
 public static void main(String args[]) {
  int a, b;
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter two nos :");
  a = sc.nextInt();
  b = sc.nextInt();
  int big;
```

```
  int small;
  if (a > b) {
   big = a;
   small = b;
  } else {
   big = b;
   small = a;
  }
  for (int i = 1; i <= big; i++) {
   if (((big * i) % small) == 0) {
    int lcm = big * i;
    System.out.println("The least common multiple is " + (lcm));
    break;
   }
  }
  int temp = 1;
  while (temp != 0) {
   temp = big % small;
   if (temp == 0) {
    System.out.println("GCD is " + small);
   } else {
    big = small;
    small = temp;
   }
  }
 }
}
```

Java program to test the Prime number.

```
import java.util.*;
class Prime {
 public static void main(String args[]) {
  int flag, x, i;
  flag = 0;
  int a[] = new int[7];
  for (x = 0; x < args.length; x++) {
   a[x] = Integer.parseInt(args[x]);
   for (i = 2; i < (a[x] / 2); i++) {
    if ((a[x] % i) == 0) {
     break;
    } else flag = 1;
   }
   if (flag == 1)
    System.out.println(a[x] + " is a prime no ");
   else
    System.out.println(a[x] + " is not a prime no ");
   flag = 0;
```

```
  }
 }
}
```

Java program to Demonstrate Type Casting.

```
class Typecast {
 public static void main(String args[]) {
  byte h = 127;
  int a = 300;
  float a1 = 12.222 f;
  float g;
  short b = 200;
  long c = 999999;
  float e = 345.89 F;
  double f = 45645.78222222222222;
  g = (float) f;
  System.out.println("short b =" + g);
  System.out.println("short b =" + b);
  System.out.println("long c =" + c);
  System.out.println("float e=" + e);
  System.out.println("double f=" + f);
  System.out.println("short b=" + b);
  System.out.println("short to byte " + (byte) b);
  System.out.println("int to byte " + (byte) a);
  System.out.println("int to float" + (float) a);
  System.out.println("long to byte " + (byte) c);
  System.out.println("double to long " + (long) f);
  System.out.println("double to int " + (int) f);
  System.out.println("double to byte " + (byte) f);
  System.out.println("double to short " + (short) f);
  System.out.println("double to float " + (float) f);
  System.out.println("float to int " + (int) e);
  System.out.println("float to byte " + (byte) e);
  System.out.println("float to short " + (short) e);
  System.out.println("float to long " + (long) e);
  System.out.println("float to double =" + (double) e);
  System.out.println("long to int" + (int) c);
  System.out.println("byte to int =" + (int) h);
 }
}
```

Find the average, sum, min and max of the N numbers Using user Input in Java.

```
import java.util.*;
class Average {
 public static void main(String args[]) {
  Scanner sc = new Scanner(System.in);    // to take user input
  int choice;
  int a = 0, min = 0, max = 0, x;
  int n = args.length;
  System.out.println("1-sum");
  System.out.println("2-Average");
  System.out.println("3-Minimum");
  System.out.println("4-Maximum");
  System.out.println("Enter Ur Choice : ");
  choice = sc.nextInt();
  for (int i = 0; i < n; i++) {
   a += Integer.parseInt(args[i]);   //to convert string into Integer
  }
  switch (choice) {
   case 1:
    System.out.println("The sum is : " + a);
    break;
   case 2:
    System.out.println("The Average is : " + a / n);
    break;
   case 3:
    for (int i = 0; i < n - 1; i++) {
     x = Integer.parseInt(args[i]);
     if (x < Integer.parseInt(args[i + 1]))
      min = x;
     else min = Integer.parseInt(args[i + 1]);
    }
    System.out.println("The minimum is : " + min);
    break;
   case 4:
    for (int i = 0; i < n - 1; i++) {
     x = Integer.parseInt(args[i]);
     if (x > Integer.parseInt(args[i + 1]))
      max = x;
     else
      max = Integer.parseInt(args[i + 1]);
    }
    System.out.println("The maximum is : " + max);
    break;
  }
 }
}
```

How can you determine if String has all Unique Characters?

```
public class DetermineDuplicate
{
   public static void main(String args[])
   {
      String s = "Umesh Kushwaha";
      int check = 0;
      for (int i = 0; i < s.length(); i++)
      {
         for (int j = 0; j < s.length(); j++)
         {
            if (s.charAt(i) == s.charAt(j) && i != j)
            {
               check = 1;
               break;
            }
         }
      }
      if (check == 1)
      {
         System.out.println("String does not have all unique character");
      } else
      {
         System.out.println("String have all unique character");
      }
   }
}
```

Output: String does not have all unique character

How to reverse a String without using any direct method.

```
public class ReverseOfString
{
   public static void main(String args[])
   {
      String s = "UMESH KUMAR";
      char a[] = new char[s.length()];
      int n = s.length() - 1;
      for (int i = 0; i < s.length(); i++)
      {
         a[i] = s.charAt(n);
         n--;
      }
      String rev = new String(a);
      System.out.println(rev);
```

```
    }
}
```

Output: RAMUK HSEMU

How to convert String to integer without using any direct method in Java.

```
public class StringToInt
{
    public static void main(String args[])
    {
        String s = "-88881";
        int a = 0;
        int num = 0;
        int check = 0;
        for (int i = 0; i < s.length(); i++)
        {
            if (s.charAt(i) == '-')
            {
                check = 1;
            } else {
                a = s.charAt(i);
                a = a - 48;
                num = num * 10 + a;
            }
        }
        if (check == 1)
        {
            num = num * (-1);
        }
        System.out.println(num);
    }

}
```

How to find the missing values from an sorted array.

```
public class MissingValue
{
    public static void main(String args[])
```

```
    {
        int[] a = {1, 3, 5, 6, 9, 50, 100};
        int prev = a[0];
        for (int i = 1; i < a.length; i++)
        {
            int gap = a[i] - prev;
            if (gap - 1 == 1) {
                System.out.printf("Missing: %d\n", prev + 1);
            } else if (gap > 1) {
                System.out.printf("Missing: %d - %d\n", prev + 1, a[i] - 1);
            }
            prev = a[i];
        }
    }
}
```

Output:

```
Missing: 2
Missing: 4
Missing: 7 - 8
Missing: 10 - 49
Missing: 51 - 99

BUILD SUCCESSFUL (total time: 0 seconds)
```

How to reverse an array without using any built in method and any other array.

```
public class ArraySorting
{
    public static void main(String args[])
    {
        int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
        int n1 = 0, n2 = a.length - 1;
        int temp = 0;
        while (n1 != n2 || n1 < n2)
        {
            temp = a[n1];
            a[n1] = a[n2];
            a[n2] = temp;
            n1++;
            n2--;
        }
```

```
        for (int i = 0; i < a.length; i++)
        {
           System.out.print(a[i] + " ");
        }
    }
}
```

Output:

9 8 7 6 5 4 3 2 1

BUILD SUCCESSFUL (total time: 0 seconds)

How to remove specific character from an String in Java.

```
public class RemoveCharacterFromArray
{
   public static void main(String args[])
   {
      String s = "Umesh kumar kushwaha";
      String r[] = s.split("s|h");
      String res = "";
      for (String m : r)
      {
         res = res + m;
      }
      System.out.println(res);
   }
}
```

Output:

Ume kumar kuwaa

BUILD SUCCESSFUL (total time: 0 seconds)

How to find the caller of a method in Java.

```
class A
{
   public void display()
   {
      StackTraceElement ste[] = Thread.currentThread().getStackTrace();
      System.out.println("method is called from " + ste[2]);
   }
}
```

```
public class GetCurrentClassName
{
   public static void main(String args[])
   {
      A a = new A();
      a.display();
   }
}
```

How to call private method from another class in java

```
import java.lang.reflect.Method;

public class A
{
   private void message()
   {
      System.out.println("hello java");
   }
}

public class MethodCall
{
   public static void main(String[] args) throws Exception
   {
      Class c = Class.forName("A");
      Object o = c.newInstance();
      Method m = c.getDeclaredMethod("message", null);
      m.setAccessible(true);
      m.invoke(o, null);
   }
}
```

JDBC Based Programs

Write a java program to create a table through frontend application?

```
import java.sql.*;
class CreateTable {
 public static void main(String[] args) {
   try {
    Class.forName("Sun.jdbc.odbc.JdbcOdbcDriver");
    System.out.println("DRIVERS LOADED...");
```

```
    Connection con = DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
    System.out.println("CONNECTION ESTABLISHED...");
    Statement st = con.createStatement();
    int i = st.executeUpdate("create table kalyan (eno number (4), ename varchar2 (15))");
    System.out.println("TABLE CREATED...");
    con.close();
   } catch (Exception e) {
    e.printStackTrace();
   }
  } // main
}  // CreateTable
```

Write a java program which illustrates the concept of Batch processing?

```
import java.sql.*;
class BatchProConcept {
 public static void main(String[] args) throws Exception {
   Class.forName("Sun.jdbc.odbc.JdbcOdbcDriver");
   System.out.println("DRIVERS LOADED...");
   Connection con = DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
   System.out.println("CONNECTION ESTABLISHED...");
   con.setAutoCommit(false);
   Statement st = con.createStatement();
   st.addBatch("insert into student values (3, 'j2ee')");
   st.addBatch("delete from student where sno=1");
   st.addBatch("update student set sname='java' where sno=2");
   int res[] = st.executeBatch();
   for (int i = 0; i < res.length; i++) {
    System.out.println("NUMBER OF ROWS EFFECTED : " + res[i]);
   }
   con.commit();
   con.rollback();
   con.close();
  } // main
}   // BatchProConcept
```

Write a java program to create a table through frontend application?

```
import java.sql.*;
class CreateTable {
 public static void main(String[] args) {
   try {
```

```
    Class.forName("Sun.jdbc.odbc.JdbcOdbcDriver");
    System.out.println("DRIVERS LOADED...");
    Connection con = DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
    System.out.println("CONNECTION ESTABLISHED...");
    Statement st = con.createStatement();
    int i = st.executeUpdate("create table kalyan (eno number (4), ename varchar2 (15))");
    System.out.println("TABLE CREATED...");
    con.close();
   } catch (Exception e) {
    e.printStackTrace();
   }
  } // main
}  // CreateTable
```

Write a java program which illustrates the concept of Batch processing?

```
import java.sql.*;
class BatchProConcept {
 public static void main(String[] args) throws Exception {
   Class.forName("Sun.jdbc.odbc.JdbcOdbcDriver");
   System.out.println("DRIVERS LOADED...");
   Connection con = DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
   System.out.println("CONNECTION ESTABLISHED...");
   con.setAutoCommit(false);
   Statement st = con.createStatement();
   st.addBatch("insert into student values (3, 'j2ee')");
   st.addBatch("delete from student where sno=1");
   st.addBatch("update student set sname='java' where sno=2");
   int res[] = st.executeBatch();
   for (int i = 0; i < res.length; i++) {
    System.out.println("NUMBER OF ROWS EFFECTED : " + res[i]);
   }
   con.commit();
   con.rollback();
   con.close();
  } // main
}   // BatchProConcept
```

Write a java program which illustrates the concept of updatable ResultSet?

```
import java.sql.*;
class UpdateResultSet {
```

```
 public static void main(String[] args) {
   try {
    Class.forName("Sun.jdbc.odbc.JdbcOdbcDriver");
    System.out.println("DRIVERS LOADED...");
    Connection con = DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
    System.out.println("CONNECTION ESTABLISHED...");
    Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
    ResultSet rs = st.executeQuery("select * from emp1");
    rs.next();
    rs.updateInt(2, 8000);
    rs.updateRow();
    System.out.println("1 ROW UPDATED...");
    rs.moveToInsertRow();
    rs.updateInt(1, 104);
    rs.updateInt(2, 2000);
    rs.insertRow();
    System.out.println("1 ROW INSERTED...");
    rs.absolute(2);
    rs.deleteRow();
    System.out.println("1 ROW DELETED...");
    con.close();
   } catch (Exception e) {
    e.printStackTrace();
   }
  } // main
}   // UpdateResultSet
```

Write a java program which illustrates the concept of scrollable ResultSet?

```
import java.sql.*;
class ScrollResultSet {
 public static void main(String[] args) {
   try {
    Class.forName("Sun.jdbc.odbc.JdbcOdbcDriver");
    System.out.println("DRIVERS LOADED...");
    Connection con = DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
    System.out.println("CONNECTION ESTABLISHED...");
    Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
    ResultSet rs = st.executeQuery("select * from emp");
    System.out.println("RECORDS IN THE TABLE...");
    while (rs.next()) {
     System.out.println(rs.getInt(1) + " " + rs.getString(2));
    }
    rs.first();
```

```
    System.out.println("FIRST RECORD...");
    System.out.println(rs.getInt(1) + " " + rs.getString(2));
    rs.absolute(3);
    System.out.println("THIRD RECORD...");
    System.out.println(rs.getInt(1) + " " + rs.getString(2));
    rs.last();
    System.out.println("LAST RECORD...");
    System.out.println(rs.getInt(1) + " " + rs.getString(2));
    rs.previous();
    rs.relative(-1);
    System.out.println("FIRST RECORD...");
    System.out.println(rs.getInt(1) + " " + rs.getString(2));
    con.close();
   } catch (Exception e) {
    System.out.println(e);
   }
  } // main
}; // ScrollResultSet
```

Write a java program which points the data of a table along with its column names?

```
import java.sql.*;
class Table {
 public static void main(String[] args) {
   try {
    DriverManager.registerDriver(new Sun.jdbc.odbc.JdbcOdbcDriver());
    System.out.println("DRIVERS LOADED...");
    Connection con = DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
    System.out.println("CONNECTION ESTABLISHED...");
    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery("select * from dept");
    ResultSetMetaData rsmd = rs.getMetaData();
    System.out.println("=========================================");
    // PRINTING COLUMN NAME
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {
     System.out.print(rsmd.getColumnName(i) + " ");
    }
    System.out.println("");
    System.out.println("=========================================");
    // PRINTING THE DATA OF THE TABLE
    while (rs.next()) {
     for (int j = 1; j <= rsmd.getColumnCount(); j++) {
      System.out.print(rs.getString(j) + " ");
     }
     System.out.println("");
    }
```

```
    con.close();
   } catch (SQLException sqle) {
    sqle.printStackTrace();
   }
  } // main
}; // Table
```

Write a java program which illustrates the concept of resource bundle file or how to develop a flexible jdbc application along with its metadata?

```
import java.sql.*;
import java.io.*;
import java.util.*;
class RBFConcept {
 public static void main(String[] args) {
   try {
    FileInputStream fis = new FileInputStream("rbfdb.prop");
    Properties p = new Properties();
    p.load(fis);
    String dname = (String) p.get("Dname");
    String url = (String) p.get("URL");
    String username = (String) p.get("Uname");
    String password = (String) p.get("Pwd");
    String tablename = (String) p.get("Tablename");
    // loading drivers and obtaining connection
    Class.forName(dname);
    System.out.println("DRIVERS LOADED...");
    Connection con = DriverManager.getConnection(url, username, password);
    System.out.println("CONNECTION CREATED...");
    // executing query
    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery("select * from" + tablename);
    ResultSetMetaData rsmd = rs.getMetaData();
    // printing column names
    System.out.println("================================");
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {
     System.out.print(rsmd.getColumnName(i) + " ");
    }
    System.out.println("");
    System.out.println("================================");
    // printing the data
    while (rs.next()) {
     for (int j = 1; j <= rsmd.getColumnCount(); j++) {
      System.out.print(rs.getString(j) + " ");
     }
    }
```

```
    con.close();
   } catch (Exception e) {
    e.printStackTrace();
   }
  } // main
}   // RSFConcept
```

Write a java program which illustrates the concept of DatabaseMetaData and ResultSetMetaData?

```
import java.sql.*;
class MetaData {
 public static void main(String[] args) {
   try {
    DriverManager.registerDriver(new Sun.jdbc.odbc.JdbcOdbcDriver());
    System.out.println("DRIVERS LOADED...");
    Connection con = DriverManager.getConnection("jdbc : odbc : oradsn", "scott", "tiger");
    System.out.println("CONNECTION ESTABLISHED...");
    // UNIVERSAL DATABASE DETAILS
    DatabaseMetaData dmd = con.getMetaData();
    System.out.println("DATABASE NAME : " + dmd.getDatabaseProductName());
    System.out.println("DATABASE VERSION : " + dmd.getDatabaseProductVersion());
    System.out.println("NAME OF THE DRIVER : " + dmd.getDriverName());
    System.out.println("VERSION OF THE DRIVER : " + dmd.getDriverVersion());
    System.out.println("MAJOR VERSION OF DRIVER : " + dmd.getDriverMajorVersion());
    System.out.println("MINOR VERSION OF DRIVER : " + dmd.getDriverMinorVersion());
    // USER DATABASE DETAILS
    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery("select * from dept");
    ResultSetMetaData rsmd = rs.getMetaData();
    System.out.println("NUMBER OF COLUMNS : " + rsmd.getColumnCount());
    for (int i = 1; i <= rsmd.getColumnCount(); i++) {
     System.out.println("NAME OF THE COLUMN : " + rsmd.getColumnName(i));
     System.out.println("TYPE OF THE COLUMN : " + rsmd.getColumnType(i));
    }
    con.close();
   } catch (Exception e) {
    e.printStackTrace();
   }
  } // main
} // MetaData
```

Write a jdbc program to retrieve the data from excel?

```
import java.sql.*;
class XSelect {
 public static void main(String[] args) {
   try {
    DriverManager.registerDriver(new
    Sun.jdbc.odbc.JdbcOdbcDriver());
    System.out.println("DRIVERS LOADED...");
    Connection con = DriverManager.getConnection("jdbc:odbc:xldsn");
    System.out.println("CONNECTION ESTABLISHED...");
    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery("select * from [student$]");
    while (rs.next()) {
     System.out.println(rs.getString(1) + " " + rs.getString(2) + " " + rs.getString(3));
    }
    con.close();
   } catch (SQLException sqle) {
    sqle.printStackTrace();
   }
  } // main
} // XSelect
```

Write a java program to retrieve the records from a specified database by accepting input from keyboard?

```
import java.sql.*;
import java.io.*;
class SelectDataRun {
 public static void main(String[] args) {
   try {
    Class.forName("Sun.jdbc.odbc.JdbcOdbcDriver");
    System.out.println("DRIVERS LOADED...");
    Connection con = DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
    System.out.println("CONNECTION ESTABLISHED...");
    PreparedStatement ps = con.prepareStatement("select * from dept where deptno");
    DataInputStream dis = new DataInputStream(System.in);
    System.out.println("ENTER DEPARTMENT NUMBER : ");
    String s1 = dis.readLine();
    int dno = Integer.parseInt(s1);
    ps.setInt(1, dno);
    ResultSet rs = ps.executeQuery();
    while (rs.next()) {
     System.out.print(rs.getString(1) + " " + rs.getString(2) + " " + rs.getString(3));
    }
    con.close();
   } catch (Exception e) {
    e.printStackTrace();
   }
```

```
  } // main
}   // SelectDataRun
```

Write a java program to insert a record in dept database by accepting the data from keyboard at runtime using dynamic queries?

```
import java.sql.*;
import java.io.*;
class InsertRecRun {
 public static void main(String[] args) {
   try {
    DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
    System.out.println("DRIVERS LOADED...");
    Connection con =
    DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:
     BudDinu "," scott "," tiger ");
     System.out.println("CONNECTION OBTAINED...");
     PreparedStatement ps = con.prepareStatement("insert into dept values
    (?,?,?)");
     DataInputStream dis = new DataInputStream(System.in);
     System.out.println("ENTER DEPARTMENT NUMBER : ");
     String s1 = dis.readLine();
     int dno = Integer.parseInt(s1);
     System.out.println("ENTER DEPARTMENT NAME : ");
     String dname = dis.readLine();
     System.out.println("ENTER LOCATION NAME : ");
     String loc = dis.readLine(); ps.setInt(1, dno);
     ps.setString(2, dname);
     ps.setString(3, loc);
     int i = ps.executeUpdate();
     System.out.println(i + "ROW(s) INSERTED...");
     con.close();
    } catch (Exception e) {
     e.printStackTrace();
    }
   } // main
  } // InsertRecRun
```

Write a java program to retrieve the data from emp database?

```
import java.sql.*;
class SelectData {
 public static void main(String[] args) throws Exception {
```

```
  DriverManager.registerDriver(new Sun.jdbc.odbc.JdbcOdbcDriver());
  System.out.println("DRIVERS LOADED...");
  Connection con = DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
  System.out.println("CONNECTION ESTABLISHED...");
  Statement st = con.createStatement();
  ResultSet rs = st.executeQuery("select * from dept");
  while (rs.next()) {
   System.out.println(rs.getString(1) + " " + rs.getString(2) + " " + rs.getString(3));
  }
  con.close();
 }
};
```

Write a jdbc program which will insert a record in the Student database?

```
import java.sql.*;
class InsertRec {
 public static void main(String[] args) {
  try {
   Driver d = new            Sun.jdbc.odbc.JdbcOdbcDriver();
   DriverManager.registerDriver(d);
   System.out.println("DRIVERS LOADED...");
   Connection con =       DriverManager.getConnection("jdbc:odbc:oradsn", "scott", "tiger");
   System.out.println("CONNECTION ESTABLISHED...");
   Statement st = con.createStatement();
   int i = st.executeUpdate("insert into student values (10,'suman',60.87);");
   System.out.println(i + " ROWS SELECTED...");
   con.close();
  } catch (Exception e) {
   System.out.println("DRIVER CLASS NOT FOUND...");
  }
 }
};
```