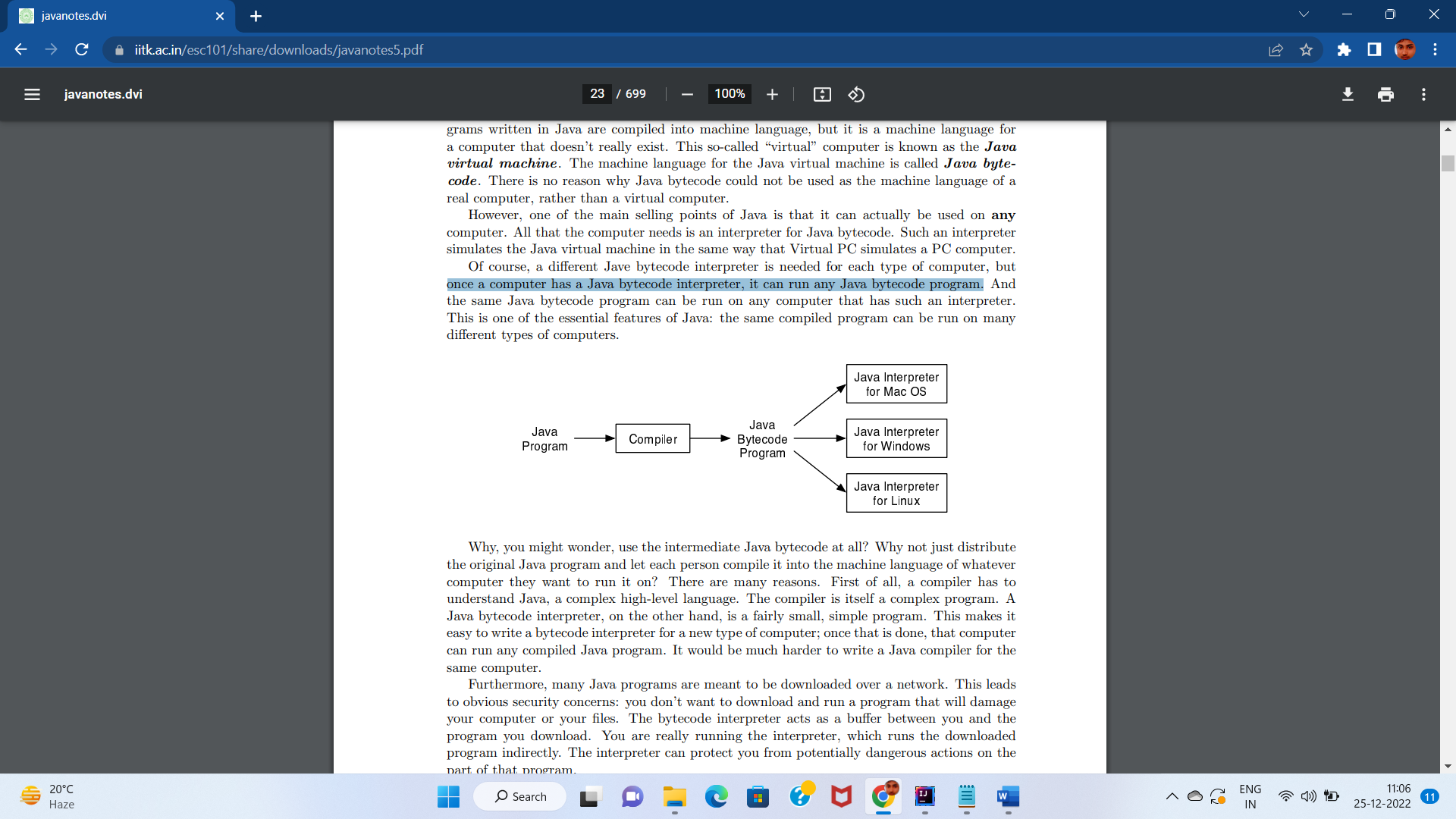
**JAVA NOTES BY SUBHASH….**

Programs written in Java are compiled into machine language, but it is a machine language for a computer that doesn’t really exist. This so-called “virtual” computer is known as the Java virtual machine. The machine language for the Java virtual machine is called Java bytecode.

once a computer has a Java bytecode interpreter, it can run any Java bytecode program.



There are two basic aspects of programming: data and instructions.

Data:-Variables , types

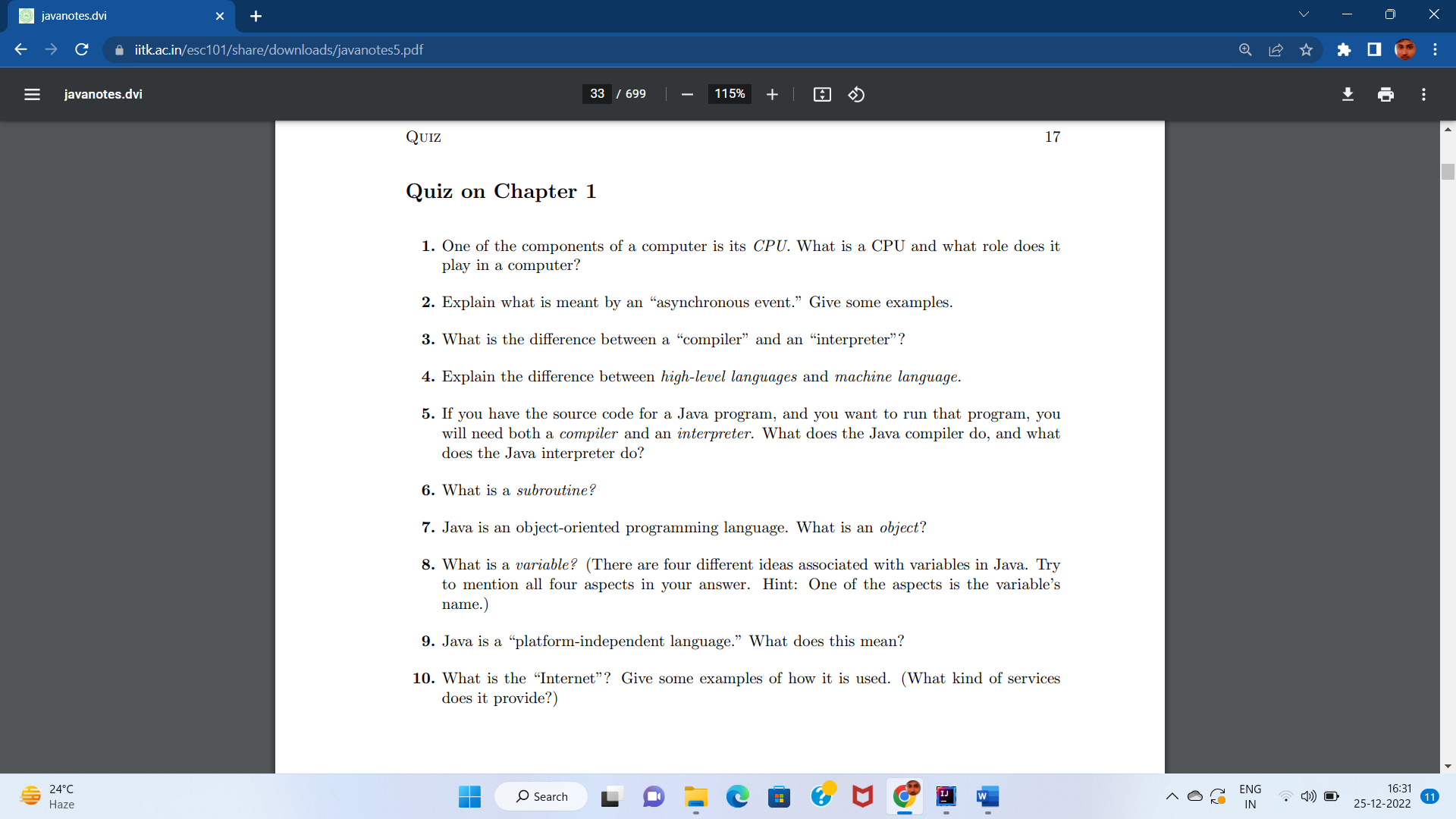
Instruction:-control structure , sub-routines

A variable is just a memory location (or several locations treated as a unit) that has been given a name so that it can be easily referred to and used in a program while a variable has a type that indicates what sort of data it can hold. Types can be integer ,floating points ,character ,strings .

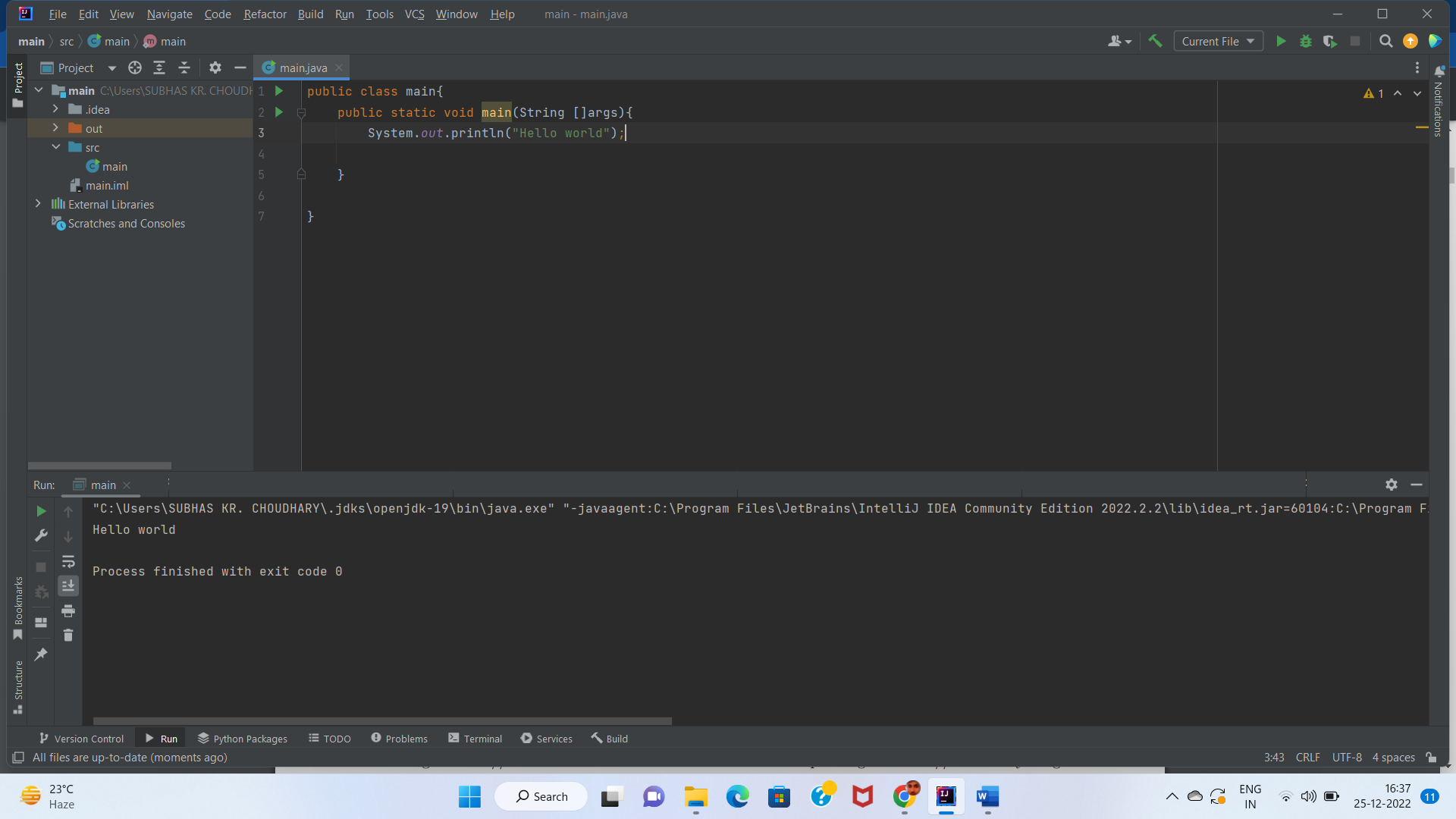
A program is a sequence of instructions. In the ordinary “flow of control,” the computer executes the instructions in the sequence in which they appear, one after the other. However, this is obviously very limited: the computer would soon run out of instructions to execute. Control structures are special instructions that can change the flow of control. There are two basic types of control structure: loops, which allow a sequence of instructions to be repeated over and over, and branches, which allow the computer to decide between two or more different courses of action by testing conditions that occur as the program is running.

A subroutine consists of the instructions for performing some task, grouped together as a unit and given a name. That name can then be used as a substitute for the whole set of instructions.

Programs must be designed. No one can just sit down at the computer and compose a program of any complexity. The discipline called software engineering is concerned with the construction of correct, working, well-written programs. The software engineer tends to use accepted and proven methods for analyzing the problem to be solved and for designing a program to solve that problem.



Answer:-



Variables and the Primitive Types

A variable is not a name for the data itself but for a location in memory that can hold data. A variable as a container or box where you can store data that you will need to use later.

The primitive types are named byte, short, int, long, float, double, char, and boolean.

The float and double types hold real numbers (such as 3.6 and -145.99).

y. A variable of type char holds a single character from the Unicode character set. And a variable of type boolean holds one of the two logical values true or false.

Any data value stored in the computer’s memory must be represented as a binary number, that is as a string of zeros and ones. A single zero or one is called a bit. A string of eight bits is called a byte.

short corresponds to two bytes (16 bits). Variables of type short have values in the range -32768 to 32767.

• int corresponds to four bytes (32 bits). Variables of type int have values in the range -2147483648 to 2147483647.

• long corresponds to eight bytes (64 bits). Variables of type long have values in the range -9223372036854775808 to 9223372036854775807.

**Fibonacci series in Java**

In fibonacci series, *next number is the sum of previous two numbers* for example 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 etc. The first two numbers of fibonacci series are 0 and 1.

There are two ways to write the fibonacci series program in java:

* Fibonacci Series without using recursion
* Fibonacci Series using recursion

## **Fibonacci Series in Java without using recursion**

public class main{  
 public static void main(  
 String [] args  
 ){  
 int n1=0,n2=1,n3,i,count=10;  
 System.*out*.print(n1+" "+n2);  
 for(i=2;i<count;i++){  
 n3 = n1+n2;  
 System.*out*.print(" "+n3);  
 n1 = n2;  
 n2 = n3;  
  
 }  
 }  
}

## **Fibonacci Series using recursion in java**

class FibonacciExample2{  
 static int *n1*=0,*n2*=1,*n3*=0;  
 static void printFibonacci(int count){  
 if(count>0){  
 *n3* = *n1* + *n2*;  
 *n1* = *n2*;  
 *n2* = *n3*;  
 System.*out*.print(" "+*n3*);  
 *printFibonacci*(count-1);  
 }  
 }  
 public static void main(String args[]){  
 int count=10;  
 System.*out*.print(*n1*+" "+*n2*);//printing 0 and 1  
 *printFibonacci*(count-2);//n-2 because 2 numbers are already printed  
 }  
}

# Prime Number Program in Java

Prime number in Java: **Prime number** is a number that is greater than 1 and divided by 1 or itself only. In other words, prime numbers can't be divided by other numbers than itself or 1. For example 2, 3, 5, 7, 11, 13, 17.... are the prime numbers.

import java.util.Scanner;  
public class main{  
 public static void main(String [] args){  
int i,m=0,flag=0;  
 System.*out*.println("enter the number: ");  
Scanner sc=new Scanner(System.*in*);  
int n=sc.nextInt();  
m=n/2;  
if(n==0|n==1){  
 System.*out*.println(n+"not prime");  
  
}  
else{  
 for(i=2;i<=m;i++){  
 if(n%i==0){  
 System.*out*.println(n+"Not prime");  
 flag=1;  
 break;  
 }  
 }  
 if(flag==0){  
 System.*out*.println(n+"prime number");  
 }  
}  
  
 }  
}

# Palindrome Program in Java

Palindrome number in java: A **palindrome number** is a number that is same after reverse. For example 545, 151, 34543, 343, 171, 48984 are the palindrome numbers. It can also be a string like LOL, MADAM etc.

## **Palindrome number algorithm**

* Get the number to check for palindrome
* Hold the number in temporary variable
* Reverse the number
* Compare the temporary number with reversed number
* If both numbers are same, print "palindrome number"
* Else print "not palindrome number"
* import java.util.Scanner;  
  public class main{  
   public static void main(String[]args){  
   int r,sum=0,temp;  
   Scanner sc=new Scanner(System.*in*);  
   System.*out*.println("Enter the number: ");  
   int n=sc.nextInt();  
   temp=n;  
   while(n>0){  
   r=n%10;  
   sum=(sum\*10)+r;  
   n=n/10;  
    
   }  
   if(temp==sum){  
   System.*out*.println("palindrome");  
   }  
   else{  
   System.*out*.println("not palindrome");  
   }  
   }  
    
  }