Compiler:

Compiler scans the entire program and translates the whole of it into machine code at once A compiler takes a lot of time to analyze the source code. However, the overall time taken to execute the process is much faster. A compiler always generates an intermediary object code. It will need further linking. Hence more memory is needed.

Interpreter:

Interpreter translates just one statement of the program at a time into machine code. An interpreter takes very less time to analyze the source code. However, the overall time to execute the process is much slower. An interpreter does not generate an intermediary code. Hence, an interpreter is highly efficient in terms of its memory.

import java.util.Scanner;

public class Student

{

private String name;

private int eng;

private int hn;

private int mts;

private double total;

private double avg;

public void accept() {

Scanner in = new Scanner(System.in);

System.out.print("Enter student name: ");

name = in.nextLine();

System.out.print("Enter marks in English: ");

eng = in.nextInt();

System.out.print("Enter marks in Hindi: ");

hn = in.nextInt();

System.out.print("Enter marks in Maths: ");

mts = in.nextInt();

}

public void compute() {

total = eng + hn + mts;

avg = total / 3.0;

}

public void display() {

System.out.println("Name: " + name);

System.out.println("Marks in English: " + eng);

System.out.println("Marks in Hindi: " + hn);

System.out.println("Marks in Maths: " + mts);

System.out.println("Total Marks: " + total);

System.out.println("Average Marks: " + avg);

}

public static void main(String args[]) {

Student obj = new Student();

obj.accept();

obj.compute();

obj.display();

}

}