|  |  |  |
| --- | --- | --- |
| *S.NO* | Compiler | Interpreter |
| *1.* | Compiler translates compiler source program in a single line. Translates the entire program. | Interpreter translate source program line by line. Translates line by line. |
| *2.* | Slow for debugging | Fast for debugging |
| *3.* | Execution time is less | Execution time is more |
| *4.* | More memory is required | Less memory is required |
| *5.* | debugging of the program is comparatively complex | error detection and correction is easy , it shows error if present at specific line |
| *6.* | Compiler generates intermediate machine code | Interpreter never produces any intermediate machine |
| *7.* | Compiler best suited for production environment | Interpreter is best suited for a software development environment |
| *8.* | Compiler used by programming languages like C,C++,Java | Interpreter is used by programming languages such as Python |
| *9.* | Execution time becomes short, code runs faster. It consumes less time | Execution is part of interpretation steps, It is slower, it’s is done line-by-line simultaneously. Interpreted programming code runs slower. It consumes more time |
| *10.* | It is more efficient | It is less efficient |
| *11.* | Compilers are larger in size | Interpreter are smaller than compilers |

Q2.

import java.util.Scanner;

public class Student {

int roll;

String name;

float marks;

void input()

{

Scanner s = new Scanner(System.in);

System.out.println("Enter roll");

roll = s.nextInt();

System.out.println("Enter Your Name");

name = s.next();

System.out.println("Enter Your Marks");

marks =s.nextFloat();

}

void display()

{

System.out.println( "Roll No is :"+roll);

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

System.out.println("Marks is :"+marks);

}

public static void main(String[] args) {

Student sd = new Student();

sd.input();

sd.display();

}

}