CT-2:

Syllabus: slide- 2.Introduce With Variables, Constants, Literals And Language Processors

Suggestions:

1. Explain Local and global variable(code)
2. How to Copy Variables? (code)
3. How to take user input of two variables and print it?
4. Calculate the Area of a Rectangle (user input).
5. What Constants and literals in C (explain with code)
6. Write two differences between compiler & interpreter?
7. Define and Draw the figure of compiler, assembler, and interpreter?

Q-1:- Explain Local and global variable(code)

```c
#include<stdio.h>

int number3 = 103; //global variable

int main(){

    //local variable

    int number1 = 101;

    int number2 = 102;

    printf("number 1 = %d\n",number1);

    printf("number 2 = %d\n",number2);

    printf("number 3 = %d\n",number3);

}
output:

number 1 = 101

number 2 = 102

number 3 = 103
```

Q-2:- How to Copy Variables? (code)

```c
#include<stdio.h>
int number3 = 103; //global variable
int main(){
    //local variable
    int number1 = 101;
    int number2 = number3;
    printf("number 1 = %d\n",number1);
    printf("number 2 = %d\n",number2);
    printf("number 3 = %d\n",number3);
}
output:
number 1 = 101
number 2 = 103
number 3 = 103
```

Q-3:-How to take user input of two variables and print it?

```c
#include<stdio.h>
int main(){
    int number1, number2;
    printf("Enter two number: ");
    scanf("%d %d", &number1, &number2);
    printf("The numbers are %d and %d\n",number1, number2);
}
output:
Enter two number: 50 60
The numbers are 50 and 60
```

Q-4:-Calculate the Area of a Rectangle (user input).

```c
#include<stdio.h>

int main() {

 // Create integer variables

 int length;

 int width;

 int area;

 printf("Enter Length : ");

 scanf("%d", & length);

 printf("Enter Width : ");

 scanf("%d", & width);

 // Calculate the area of a rectangle

 area = length * width;

 // Print the variables

 printf("Area of the rectangle is: %d\n\n\n", area);

}
```

```
Output:

Enter Length : 12

Enter Width : 20

Area of the rectangle is: 240
```

Q-5:- What Constants and literals in C (explain with code)

constants: If you don't want others (or yourself) to change existing variable values, you can use the const keyword. This will declare the variable as "constant", which means unchangeable and read-only.

literals: In C, Literals are the constant values that are assigned to the variables. Literals represent fixed values that cannot be modified.

Constants

| Const	int	var	=	5;
Keyword	Data type	Name of Student		literals
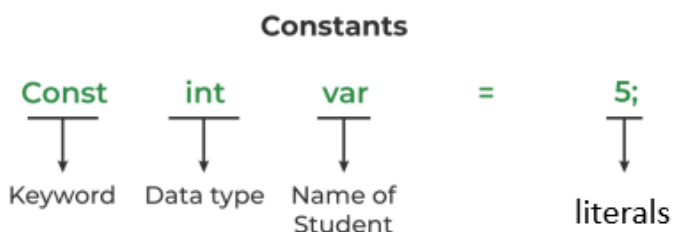
```c
#include <stdio.h>

int main(){

// constant integer literal

const int intVal = 10;

printf("Integer Literal:%d \n", intVal);

return 0;

}
```

## Q-6:- Write two differences between compiler & interpreter?

Compiler	Interpreter
A compiler is a program that converts the entire source code of a programming language into executable machine code for a CPU.	An interpreter takes a source program and runs it line by line, translating each line as it comes to it.
The compiler takes a large amount of time to analyze the entire source code but the overall execution time of the program is comparatively faster.	An interpreter takes less amount of time to analyze the source code but the overall execution time of the program is slower.

Q-7:- Define and Draw the figure of compiler, assembler, and interpreter?

Compiler: In a compiler, the source code is translated to object code successfully if it is free of errors.
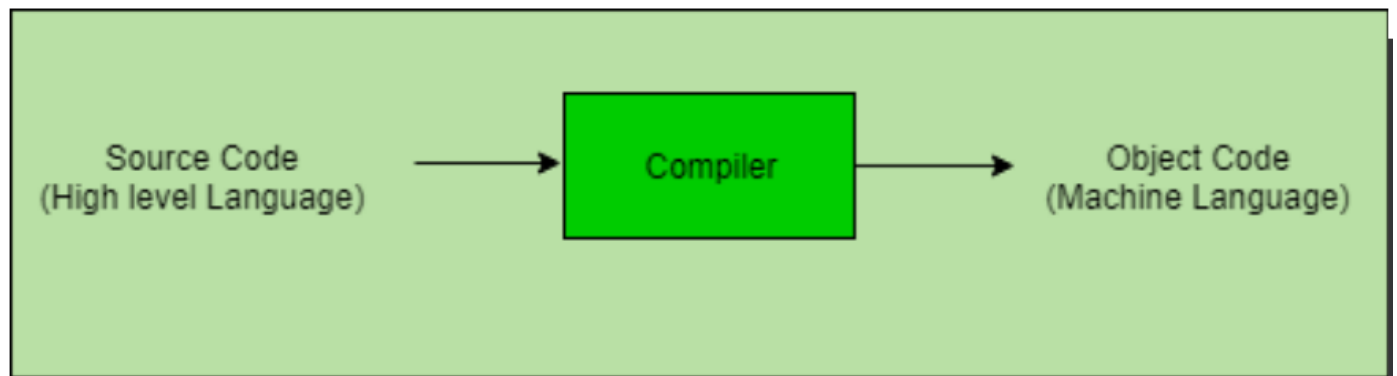
Assembler: The Assembler is used to translate the program written in Assembly language into machine code
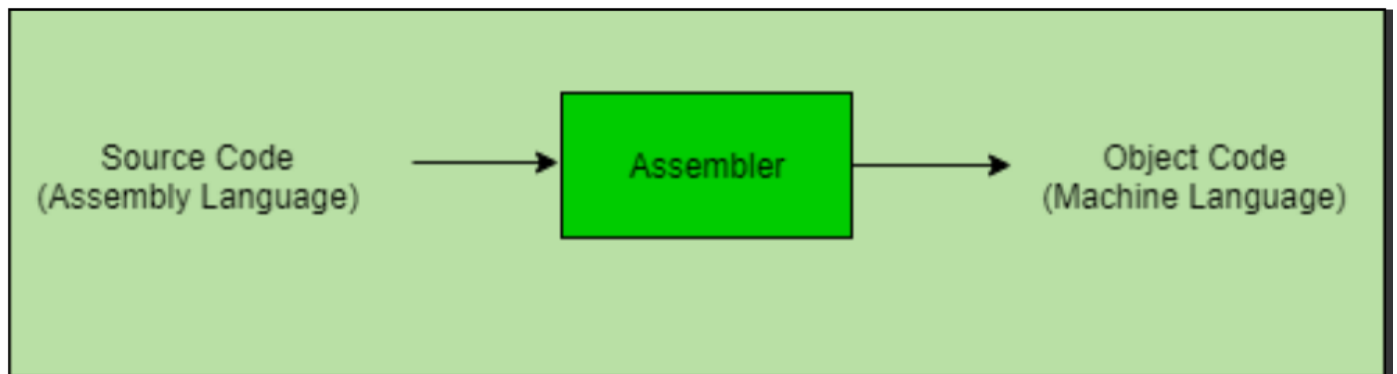
Interpreter: The translation of a single statement of the source program into machine code is done by a language processor and executes immediately before moving on to the next line is called an interpreter.