

INTRODUCTION TO C

C is a

- 1. It is a high level / middle level programming language.**
- 2. C is a compiler based programming language.**
- 3. C is a procedure oriented programming language[POP's].**
- 4. C is a general purpose programming language.**

What is a program?

Set of instructions is called program.

What is a software?

Set of programs is called software. As per it industry software is a digitalized and automated process.

```
/* source code / source program */  
#include<stdio.h>  
void main()  
{  
int i; /* variable */  
for(i=1;i<=100;i++)  
{  
printf("",i);  
}  
}
```

computer programming languages like
C / C++ / Java / .net / python



set of instructions



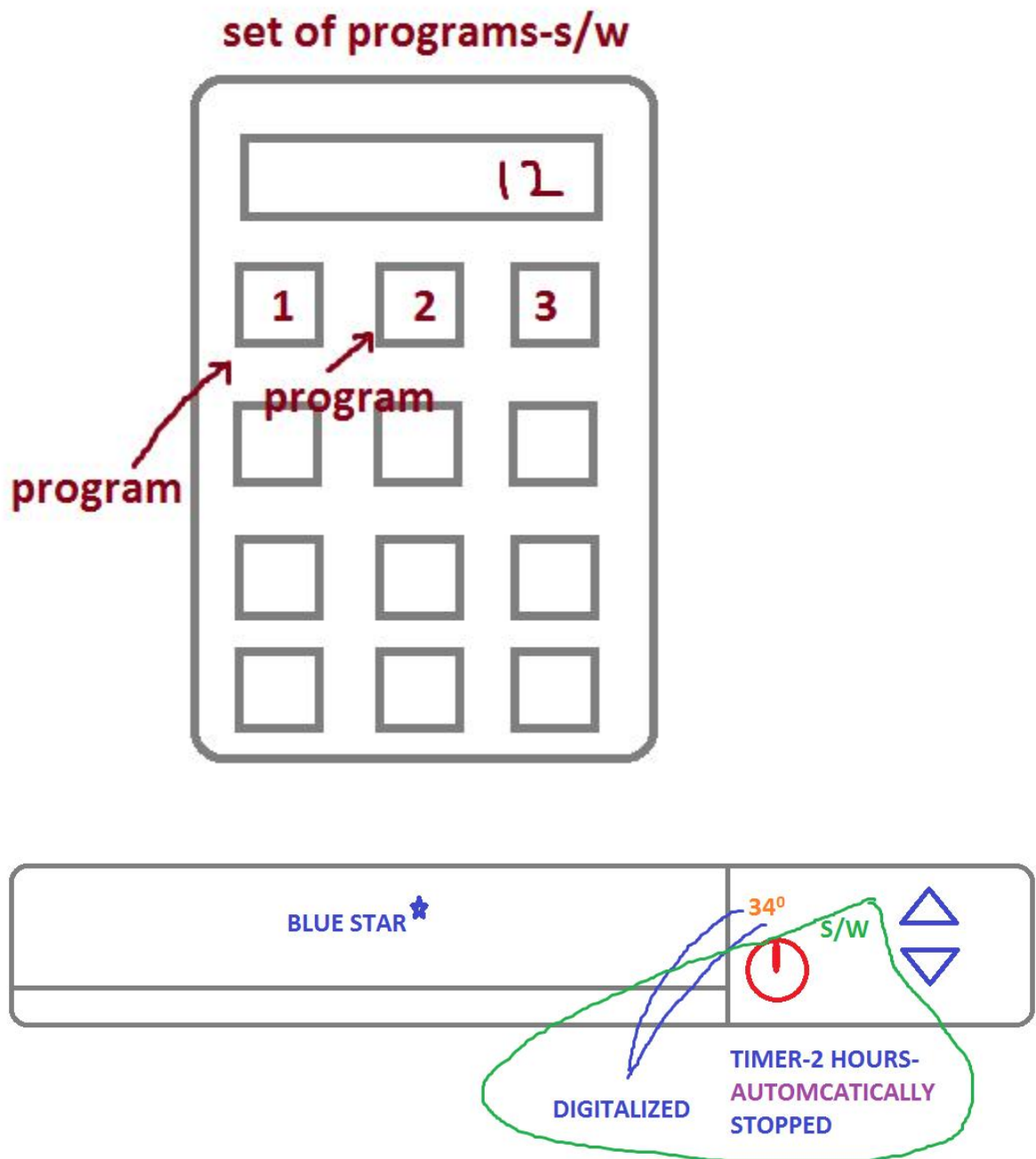
set of programs



software



tcs/wipro/microsoft - software companies - software engineer/developers/programmers/techies



We are having 2 type of software.

1. System software

Eg: o.s, device drivers, translators

2. Application software

Eg: phonepe, irctc, whatsapp,instagram,...

What is a language?

Generally the languages like telugu / English / Marathi / hindi etc are used to communicate with humans. Hence they are called human languages / regional languages. By using these human languages we can't communicate with the machines. Hence we are using the computer programming languages like C / C++ / Java / .net / pythos etc. By using these languages we are creating the **software** [programs] to communicate.

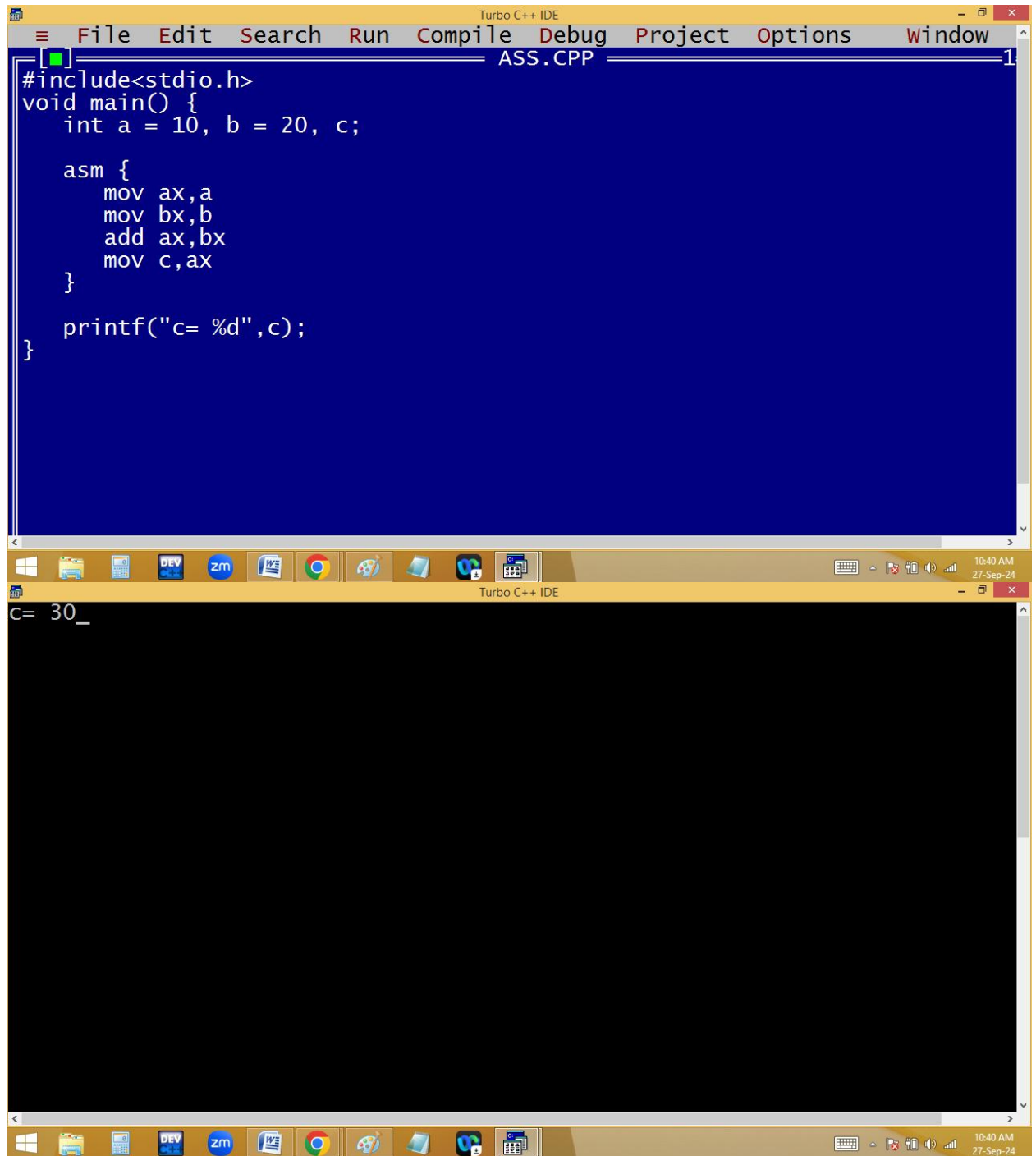
Basically the languages are divided into 3 types.

1. **Machine language**: Created with **binary code** [0,1] and very difficult to understand.

Eg: 1000111

2. **Low level / assembly language**: Created with English like shortcuts called **MNEMONICS**.

Eg: gd mrg, plz, sub,.....



The image shows a screenshot of the Turbo C++ IDE. The top window, titled 'ASS.CPP', contains the following C code:

```
#include<stdio.h>
void main() {
    int a = 10, b = 20, c;

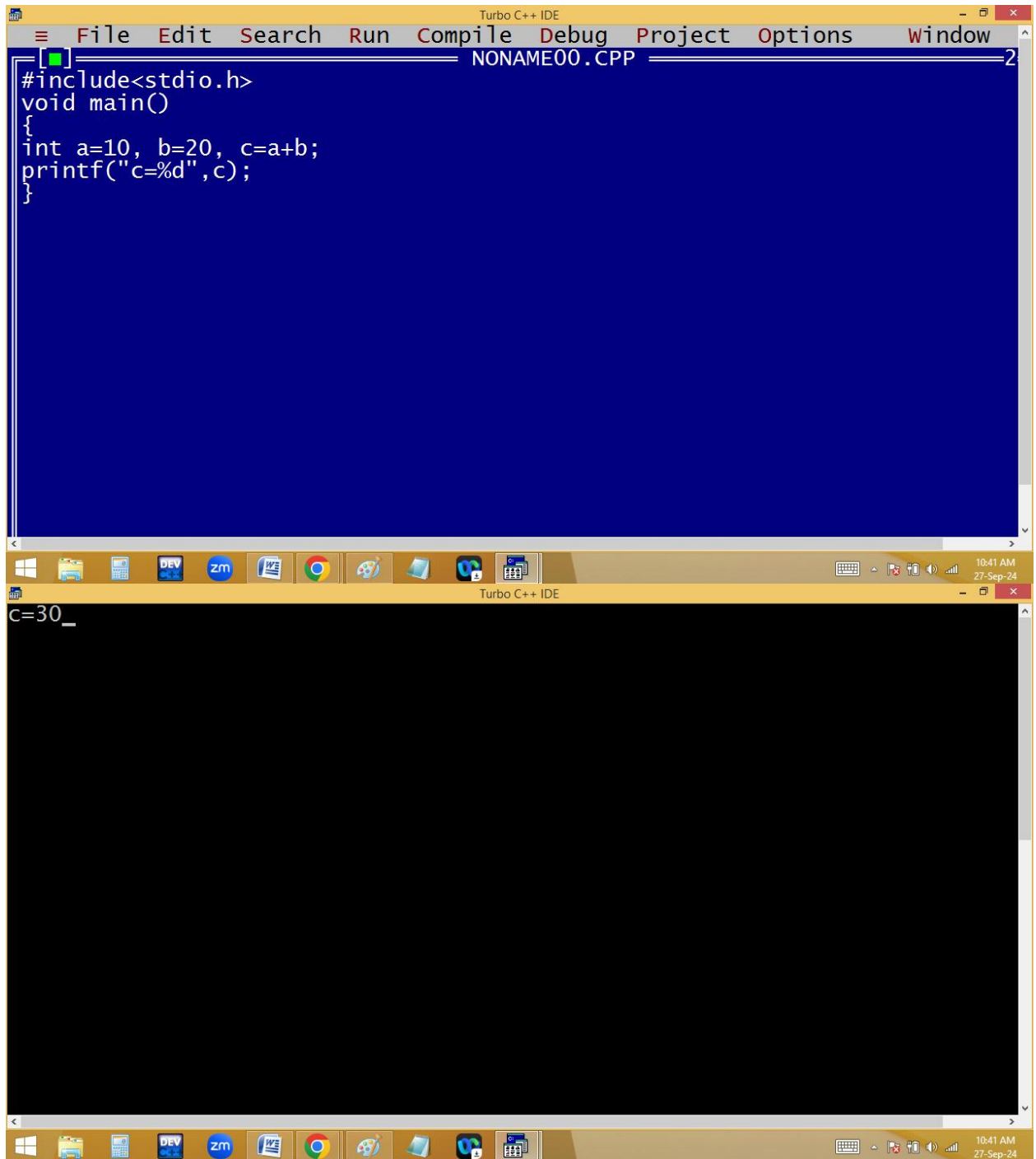
    asm {
        mov ax,a
        mov bx,b
        add ax,bx
        mov c,ax
    }

    printf("c= %d",c);
}
```

The bottom window shows the output of the program: 'c= 30_'. The Windows taskbar at the bottom includes icons for various applications and the system clock, which shows 10:40 AM on 27-Sep-24.

3. High level language: Created with **simple English** and easy to understand.

Eg: good morning, please, subject,....



The image shows two windows of the Turbo C++ IDE. The top window, titled 'NONAME00.CPP', contains the following C code:

```
#include<stdio.h>
void main()
{
int a=10, b=20, c=a+b;
printf("c=%d",c);
}
```

The bottom window shows the output of the program: 'c=30_'. The Windows taskbar at the bottom indicates the time is 10:41 AM on 27-Sep-24.

C comes with both **low level and high level features**. Hence it is a middle level language. Because of both features using c we can develop **system software and**

application software. Hence **c** is a **multi-purpose** programming language.

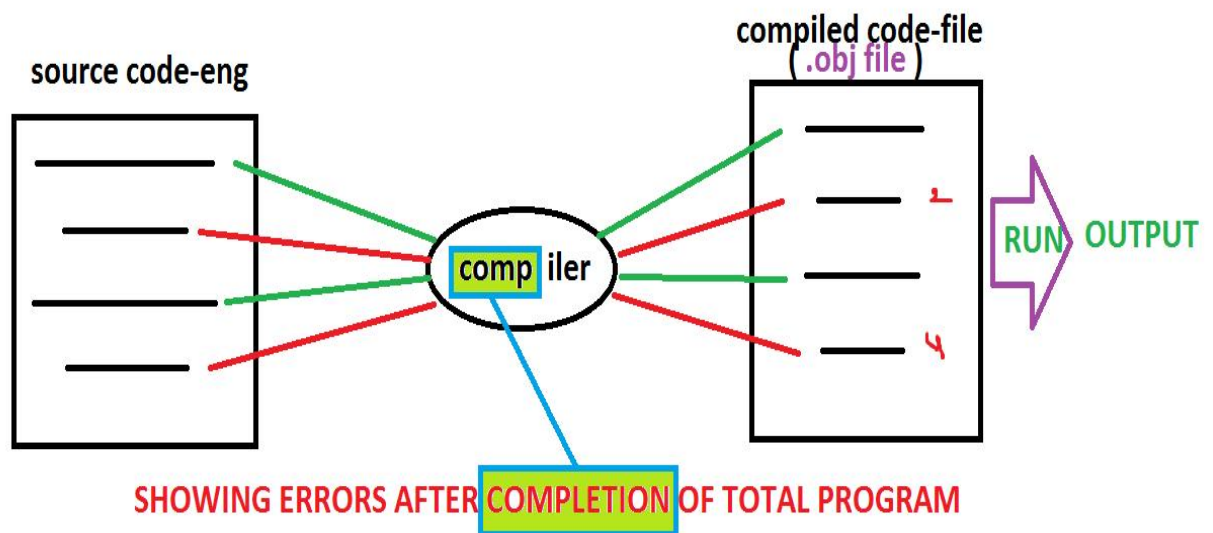
What is a compiler?



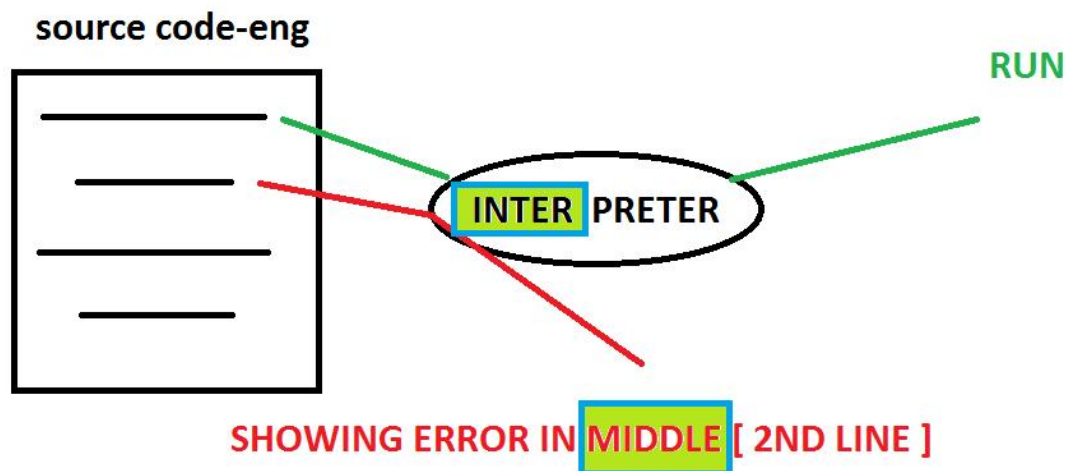
Always the user given instructions are in English, which is called source code or source program. But the machine understandable language is binary code / machine language. **To convert the source code to binary code and to check the errors we are using the translators like Compiler / interpreter / assembler.**

Compiler and interpreter both are used to convert high level programs to machine language [binary code].

Compiler converts the total program into binary code at once by leaving error lines.



Interpreter converts line by line.



Assembler is used to convert low level programs to binary code.

Assembler working style is similar to the compiler.

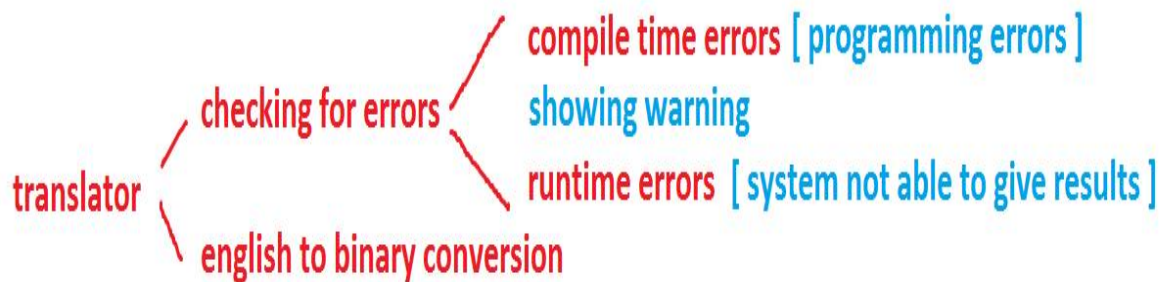
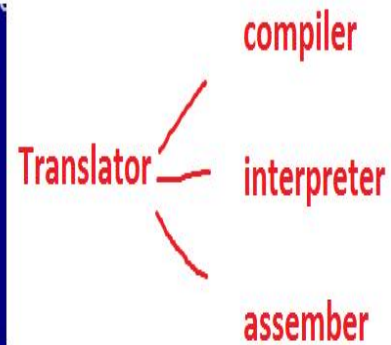
In c language we are using compiler as a interpreter. Hence it is a compiler based programming language.

In java we are using both compiler and interpreter. Hence it is a compiler based interpreted language.

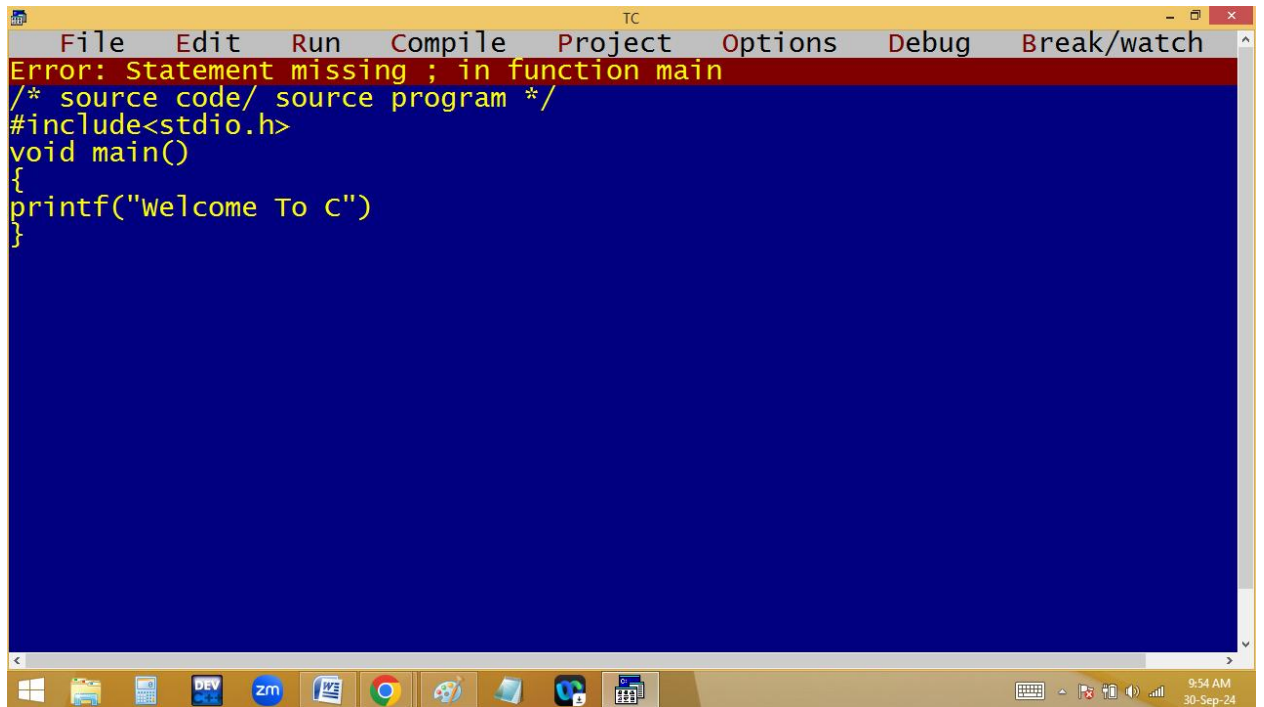

```

/* source code/ source program */_
#include<stdio.h>
void main()
{
printf("Welcome To C");
}

```



Example for compile time error:

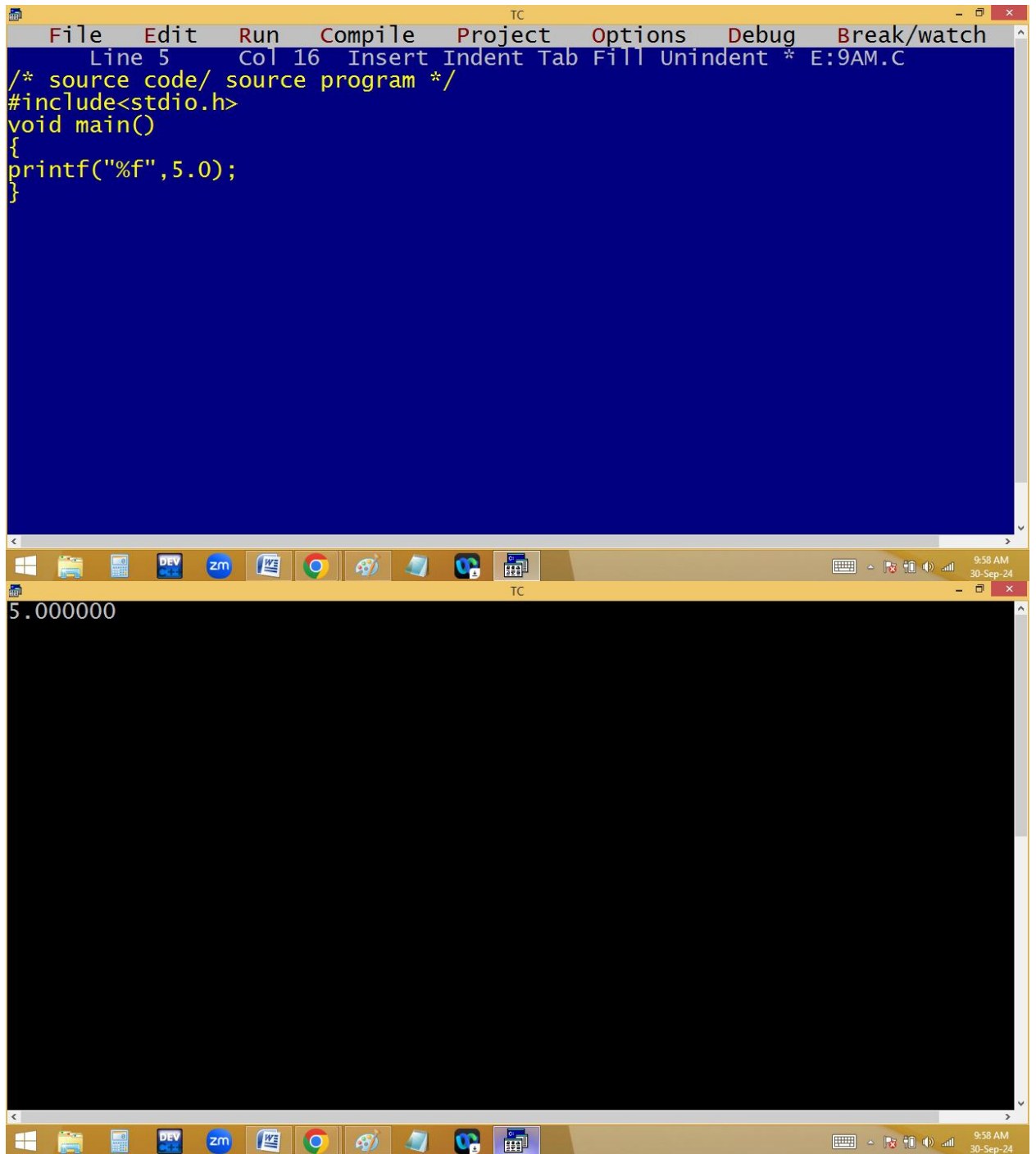


The image shows a screenshot of the Turbo C++ (TC) IDE. The window title is "TC". The menu bar includes "File", "Edit", "Run", "Compile", "Project", "Options", "Debug", and "Break/watch". The main editing area has a blue background and displays the following C code:

```
Error: Statement missing ; in function main
/* source code/ source program */
#include<stdio.h>
void main()
{
printf("Welcome To C")
}
```

The error message "Error: Statement missing ; in function main" is displayed at the top of the code area. The code is missing a semicolon at the end of the `printf` statement. The Windows taskbar is visible at the bottom, showing icons for various applications and the system clock indicating 9:54 AM on 30-Sep-24.

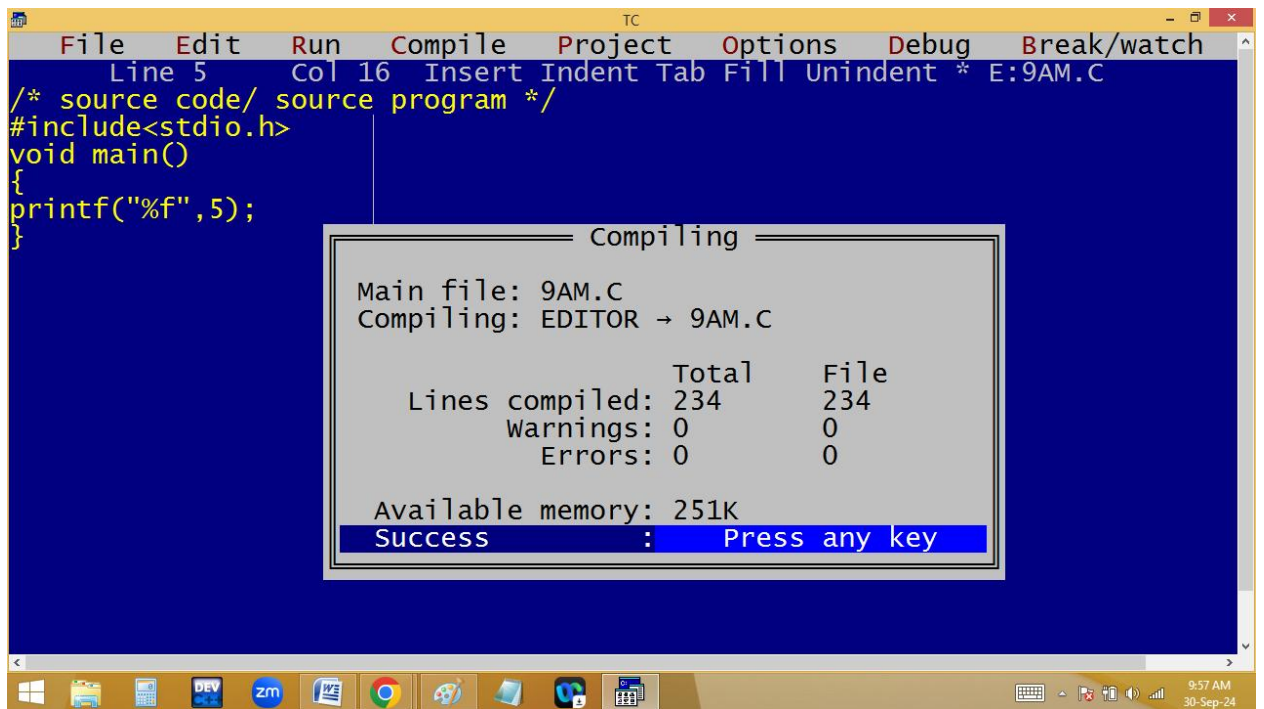
Example for runtime error:

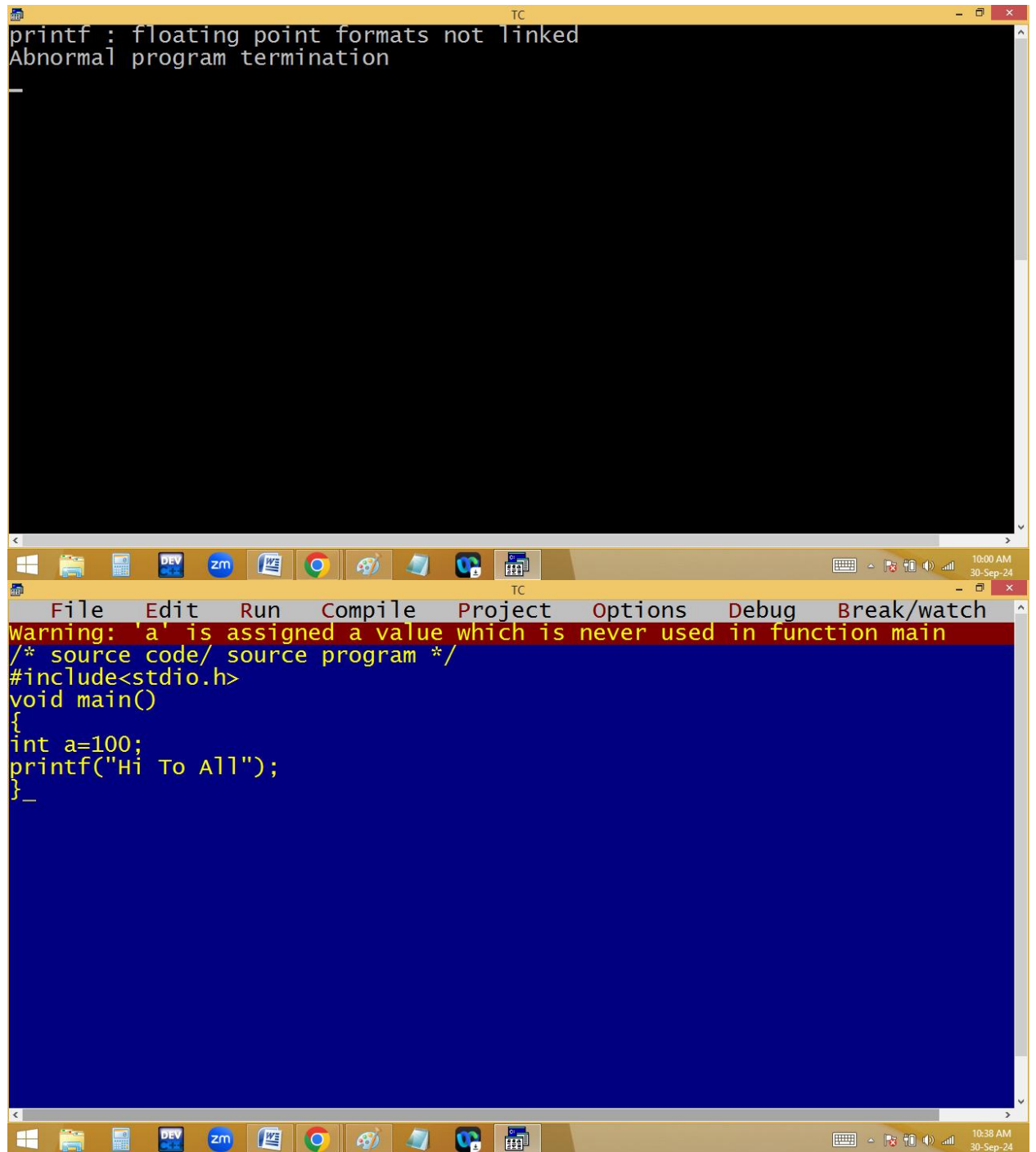


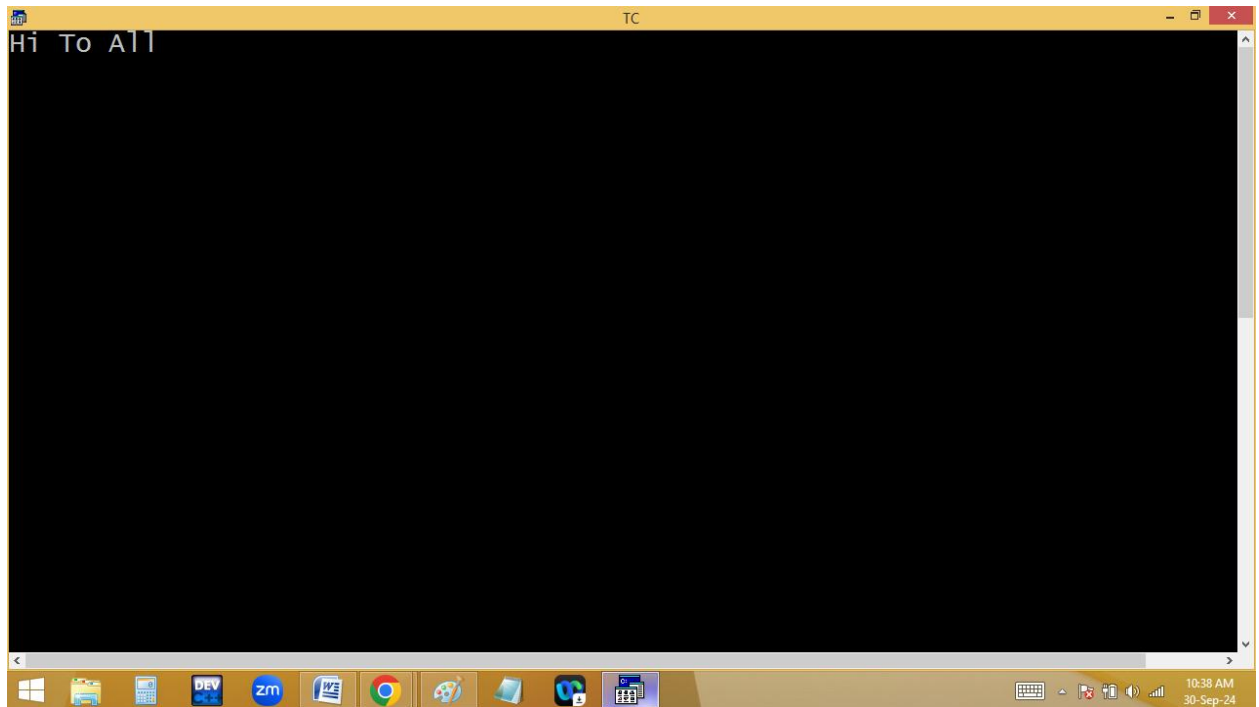
The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the source code editor, which has a blue background. It contains a C program that includes `<stdio.h>` and has a `main` function that prints the floating-point number `5.0` using `printf("%f", 5.0);`. The status bar at the top of this window indicates 'Line 5', 'Col 16', and the filename 'E:9AM.C'. Below the code editor is a black output window. The first line of the output is '5.000000', which is the result of the `printf` statement. The Windows taskbar is visible at the bottom of the screen, showing various application icons and the system clock indicating 9:58 AM on 30-Sep-24.

```
File Edit Run Compile Project Options Debug Break/watch
Line 5 Col 16 Insert Indent Tab Fill Unindent * E:9AM.C
/* source code/ source program */
#include<stdio.h>
void main()
{
printf("%f",5.0);
}
```

5.000000





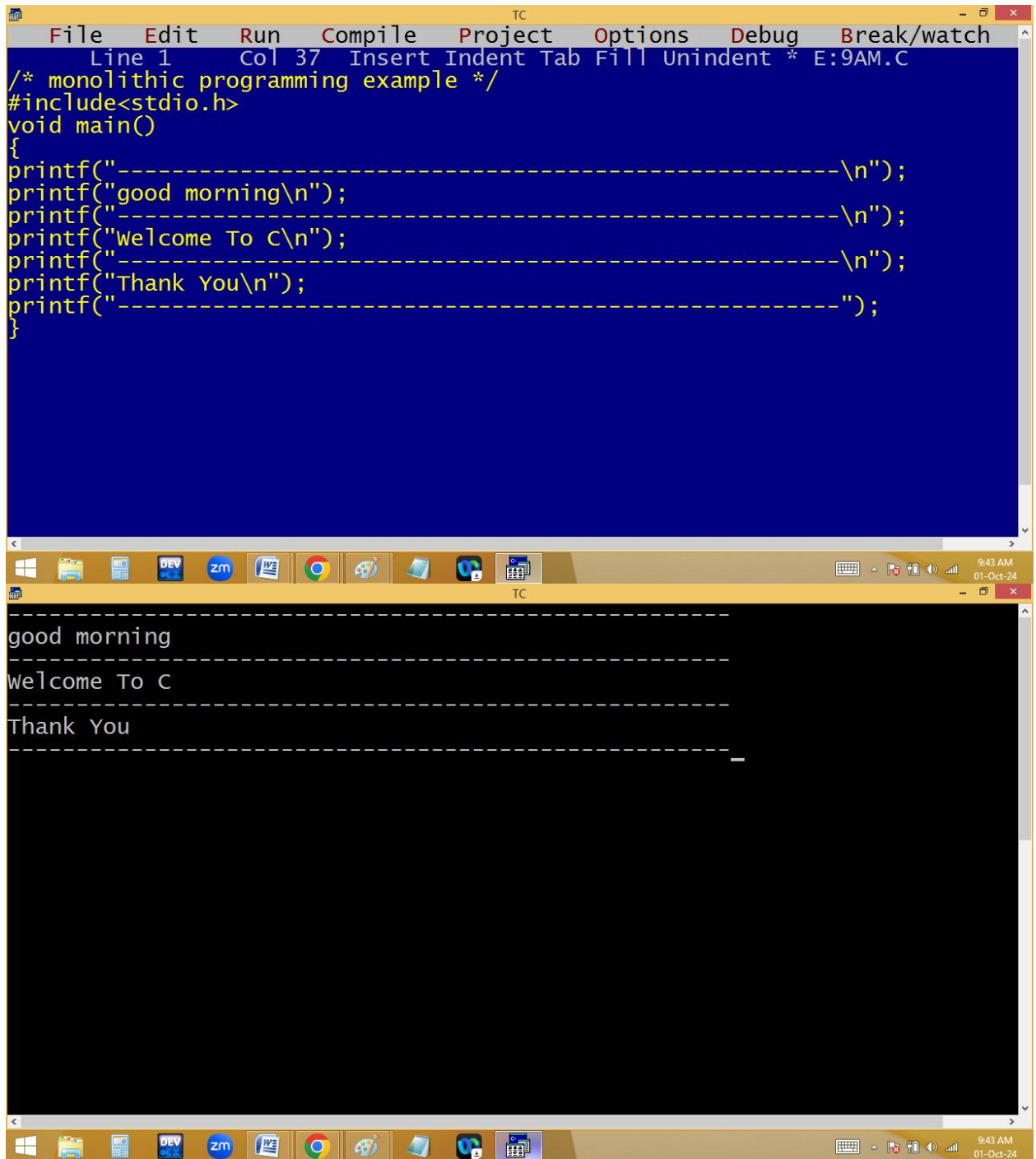


What is called programming paradigm?

Every programming language comes with certain rules and regulations with a particular structure, which is technically called programming paradigm.

Before C language, the languages are using **monolithic programming paradigm**. Here the entire program they are creating by using a single program. Due to this it is very difficult to

1. Find the errors
2. Take more memory
3. Performance is low
4. No reusability
5. Program size increased



The image shows a screenshot of a Turbo C++ (TC) IDE. The top window displays a C program with the following code:

```
File Edit Run Compile Project Options Debug Break/watch
Line 1 Col 37 Insert Indent Tab Fill Unindent * E:9AM.C
/* monolithic programming example */
#include<stdio.h>
void main()
{
printf("-----\n");
printf("good morning\n");
printf("-----\n");
printf("welcome To C\n");
printf("-----\n");
printf("Thank You\n");
printf("-----");
}
```

The bottom window shows the output of the program:

```
-----
good morning
-----
Welcome To C
-----
Thank You
-----
```

The Windows taskbar at the bottom shows the time as 9:43 AM on 01-Oct-24.

What is procedure oriented programming structure[POP's]: To avoid the drawbacks in monolithic programming problems, in C they have divided a big program into several small sub programs / sub routines /

procedures / functions / modules / structures. i.e. C program is collection of procedures, it is called POP's.

POP's Advantages:

1. **Modularity**: Dividing big program into several small pieces as per the project requirement.
2. **Simplicity**: easy to read and understand.
3. **Reusability**: Write once, use many times.
4. **Efficiency**: Performance is high.

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays the source code of a C program. The code defines a function `line()` that prints a dashed line, and a `main()` function that calls `line()` and prints "good morning", "Welcome To C", and "Thank You". The bottom window shows the output of the program, which matches the printed text in the `main()` function, with dashed lines separating the output lines. The Windows taskbar at the bottom shows the time as 10:12 AM on 01-Oct-24.

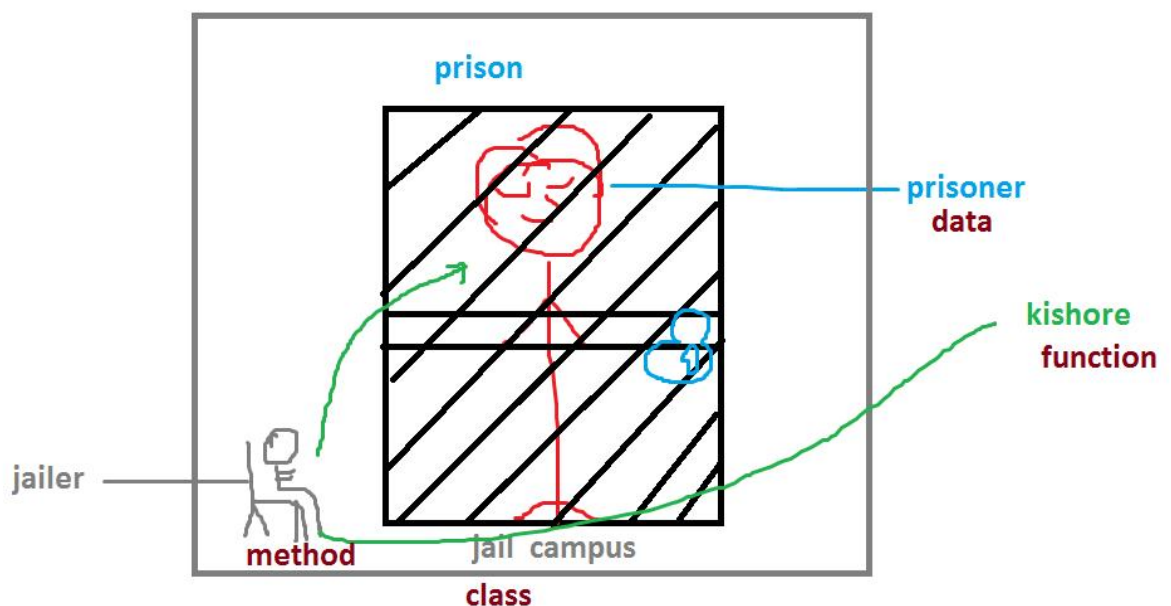
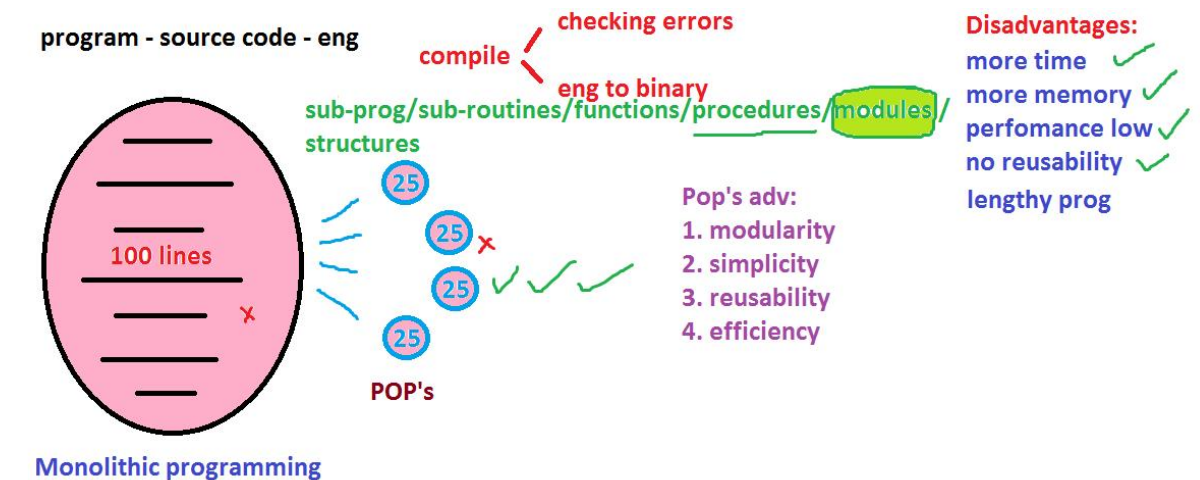
```
File Edit Run Compile Project Options Debug Break/watch
Line 1 Col 44 Insert Indent Tab Fill Unindent * E:9AM.C
/* Procedure oriented programming structure_*/
#include<stdio.h>
void line()
{
printf("-----\n");
}

void main()
{
line();
printf("good morning\n");
line();
printf("Welcome To C\n");
line();
printf("Thank You\n");
line();
}
```

good morning

Welcome To C

Thank You



POP's Disadvantages:

In C the data is not secured because of by default it is **public**.

OOP's:

Object Oriented Programming Structure

Features:

- 1. Class – It is a blue print to create the objects.**
- 2. Object: it is the physical instance/copy of class**
- 3. Data hiding – private / public / protected**
- 4. Encapsulation**
- 5. Inheritance**
- 6. Poly-many morphism-shapes / kinds / forms**
- 7. Abstraction - briefing**

class father – programmer – super / base class

{

200 lines

2 crore building;

};

Class newclass create from oldclass / sub / derived class

{

200 lines;

Class child derived from father

{

2 crore building

```
}
```

Class stu

```
{
```

```
...;
```

```
...;
```

```
}
```

Class emp

```
{
```

```
...;
```

```
...;
```

```
}
```

Why c is a general purpose language?

Using c language we can develop the software like

- 1. Operating systems**
- 2. Device drivers**
- 3. Translators**
- 4. Commercial applications**

5. Editors

6. Data base

7. Embedded applications

8. Antivirus

9. Browsers

10. Media players

11. PC & Mobile games

12. Any type of standalone applications