2

CHAPTER

Languages and Translators

Objective

When a particular problem is to be solved, it is necessary to design the instructions for computer to carry out and the art of writing instructions for a computer to solve the specific task is known as language and translator. Computer understands the instructions written in machine language. Thus, the program written in any other language needs to be translated into machine language. This task is done by translators which are special programs to check user programs grammatically and produce a corresponding set of machine language.

2.1 LANGUAGE AND TRANSLATOR

This chapter on programming will help the users to make the programs. In this section we will enquire, why do we use the language and what are the roles of computer language. This chapter contains all steps by which user will be able to understand about data type, translator and steps to write and run C++ programs.

2.2 LANGUAGE OF COMPUTER

- It is a medium by which a computer can communicate with the machine.
- Language of computer is used to control the behavior/activity of the machine.
- In computer science and engineering, human language is different from computer languages.
- Unfortunately, computer cannot understand human languages.
- Computers use special language such as C++ language, etc.
- We must communicate with computer using special languages.

(b) **Assembly language:**

- It is also known as a symbolic language.
- In assembly language specified symbols are used.
- It is a low level language.
- Easier to correct the codes.
- Easy to modify the instructions.
- It has a demerit that it is machine dependent.

2.4 TRANSLATOR

- Translator is used for translating the input language to output language.
- It is used for converting/changing the high level language or low level language into machine level language.
- It is a software that is used to convert the programming language to machine language without losing anything.

Translator are classified into three types:

(a) Compiler

(b) Interpreter

(c) Assembler

(a) **Compiler:**

- Compiler is a translator.
- It translates the source program into machine equivalent code.
- It reads the entire source program and converts it into object code and also lists the errors.

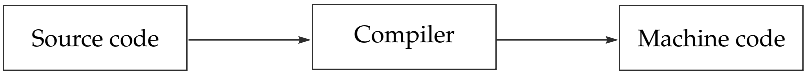


Fig. 2.1 Compiler

- Large amount of time is taken by a compiler to process high level languages.

(b) **Interpreter:**

- Interpreter is also a translator.
- It translates the source program into machine equivalent code.
- It is used to read the source program line by line or statement by statement.
- It is not generated by the intermediate object code.

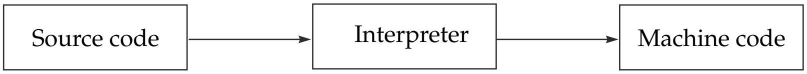
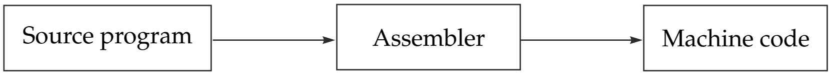


Fig. 2.2 Interpreter

(c) **Assembler:**

- Supports machine instructions in mnemonic form.
- It is machine oriented language.
- It follow the machine instructions.
- It uses the English words such MOVE, ADD, SUB, etc.
- It is used to convert the assembly language into machine language/code.



(d) **Linker**: Tool that merges the object files produced by separate compilation or assembly, and creates an executable files.

Tasks of linker:

1. Searches the program to find library routine used by program such as "cout <<" and printf( );
2. Determine the memory locations that code form each module will occupy and reallocate its instructions by adjusting absolute references.
3. Resolves references among files.

(e) **Loader**: Part of the operating system that brings an executable files residing on disk into memory and starts it running.

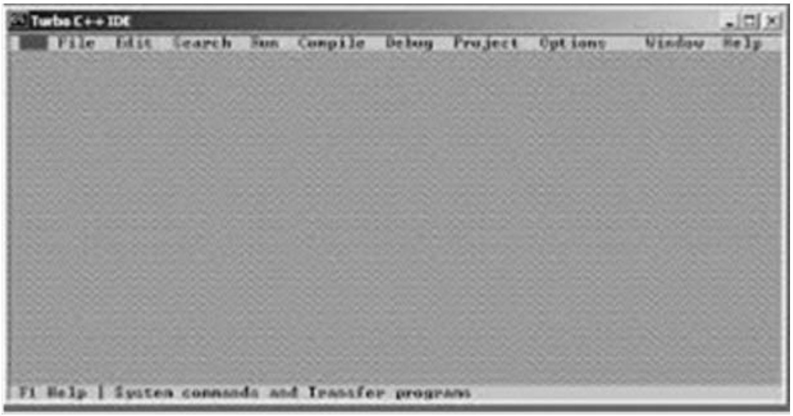
Steps of Loader

1. Read executable file's header to determine the size of text and data segments.
2. Create a new address space for the program.
3. Copies instructions and data into address space.
4. Copies arguments passed to the program on the stack.
5. Initializes the machine registers including the stack pointer.
6. Jumps to a startup routine that copies the program's arguments from the stack to registers and calls the program's main routine.

- These two parts must appear between the opening braces and closing braces.
- The program execution begins at the opening brace and end at the closing brace.
- The closing brace of the main() function section is the logical end of the program.
- Every statement in between the opening and closing braces must end with (;).

2.7 STEP-BY-STEP EXECUTION OF C++ PROGRAM

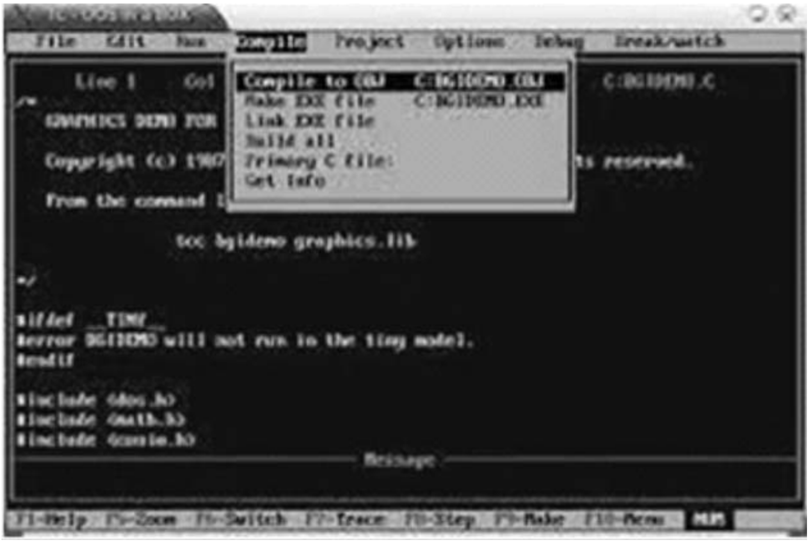
Step 1: Open the turbo screen and then write the program on turbo c/c++ screen.



F1 Help | System commands and Transfer programs

Step 2: Save this program after pressing F2 key.

Step 3: Compile the program to check and list the errors if any(press alt F9).



```
/*
   GRAPHICS DEMO FOR
   Copyright (c) 1987                          ts reserved.
   From the command l
              tcc bgidemo graphics.lib
*/
#ifdef __TINY__
#error BGIDEMO will not run in the tiny model.
#endif

#include <dos.h>
#include <math.h>
#include <conio.h>
```

Step 4: If any error occurs then remove these errors.

Step 5: Run the program (press ctrl F9).

Step 6: To show the output/console screen (press alt F5).

Terminology of ‘C++’:

1. **# (Hash) –** Hash is a symbol which is used for special purpose.
2. **include –** It is used to combine or join or to link.
3. **#include –** It is a preprocessor. #include<file name> does not search the path because it fixed in (tc-bin) and computer know where source directory is. Such as #include<iostream.h>

   Where #include< > header file format.

Header file:

- It is a preprocessor directive.
- It is an instruction to the compiler to make it do something.
- It tells the ‘C++’ compiler to include the contents of a file, respectively.

(a) **#include "file name":** #include “file name” searches the source path first, then include path. For example: #include “math.h”

(b) **#include < >**

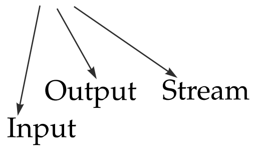
Angle bracket (search bracket)

(c) **#include**

Preprocessor

Preprocessor is used for path identification of the source directory.

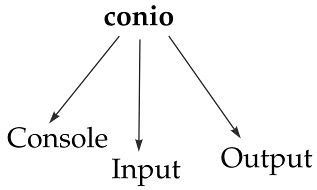
(d) **iostream**



It stands for stream buffer memory for input output device.

Other examples of header files

#include<conio.h>



```
void main( )
{

}
```

Or

```
int main( )
{

return 0;
}
```

It is used for termination of the program.

Where:

a. **void (type):** Void means that the function does not return value.

b. **main( ):** main is the first thread or process that executed by compiler. When compiler starts executing the program, only main function is available that do not need any other object to be called (actually it is static that can be called by compiler directly) but other function does not that. `It is the user defined function.`_main( ) is a primary function which is included all C++ programs. When we write `void main()`, it means that this main( ) program will not return value and if we write `int main()`, it means this program will return an integer type value. `Main is also used for entry point of program.`

Console input and output functions:

The two standard C++ programming language functions for console input and output are

1. **cout:** It stands for console output which is used to print the information or message from user or system. It uses the insertion operator(<<) to access the data/information.

   Syntax of cout:

   ```
   cout<< expression;
   ```

   programming example of cout:

   ```
   cout<< a;
   ```

   This will print the value of a.

2. **cin:** It stands for the console input which is used to read the data or information and stored it into memory places. It uses the extraction operator(>>) to access the data/information.

Syntax of cin:

```
cin>>variable_name;
```

programming example of cin:

```
cin>>a;
```

It means ‘a’ is the variable that is being scanned (read) and stored (create the memory for ‘a’) in memory.

Terminology can be summarized as:

```
#include<iostream.h>  }
#include<conio.h>     }  Header file section

void main( )   ◄──────────────  Function name
{              ◄──────────────  start of program
……  …          ◄──────────────  Program statement
}              ◄──────────────  end of program
```

My first program of ‘C++’:

```
#include<iostream.h>
#include<conio.h>
void main( )
{
clrscr( );
printf(“hello”);
getch( );
}
Output:
hello
```

Data Types:

1. Data type is a list of words that is predefined in the compiler.
2. Data type is a type or reorganization of data/variables.
3. Data type is a container/storage that holds the specific type and range of values.
4. Each variables/data must have data type.
5. Basically, it is a predefined user dictionary prepared by the compiler.

8 bits = 2^8 = 256

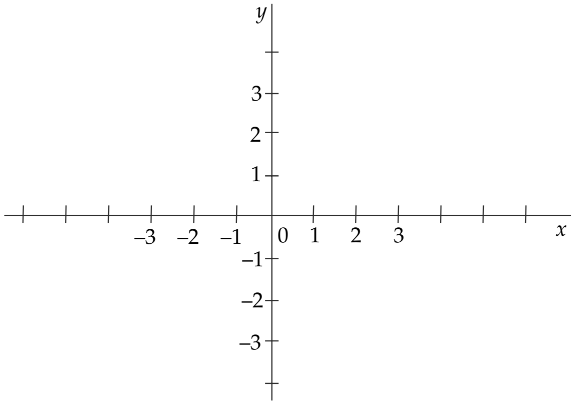
16 bits = 2^{16} = 65536

32 bits = 2^{32} = 4294967296

…… similarly rest of the data type would be calculated in bits.

Range calculation for character:

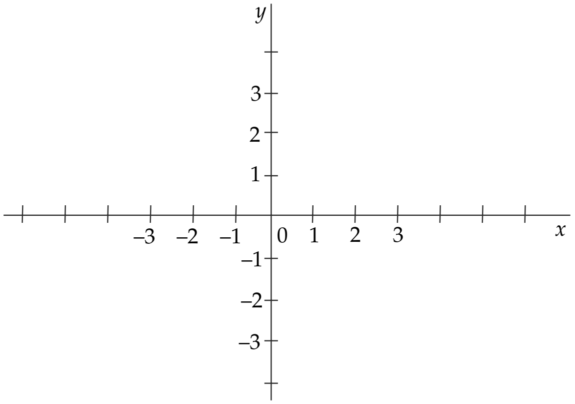
256/2=128



Therefore, the range will be (-128 to 127)

Range calculation for integer:

65536/2=32768

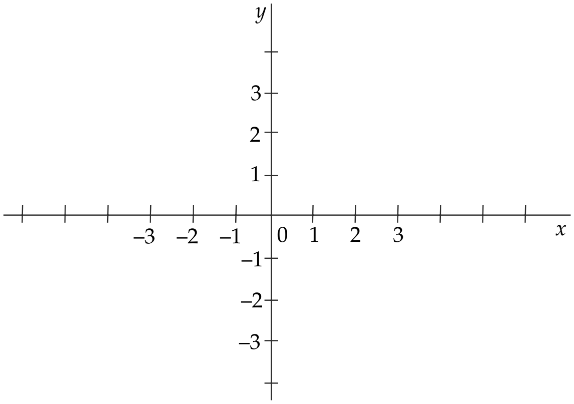


Therefore, the range will be (-32768 to 32767)

Type	Size (in bytes)	Range
int	2	– 32,768 to 32,767
signed int	2	– 32,768 to 32,767
unsigned int	2	0 to 65,535
short int	2	– 32,768 to 32,767
signed short int	2	– 32,768 to 32,767
unsigned short int	2	0 to 65,535
long int	4	– 2,147,483,648 to 2,147,483,647
signed long int	4	– 2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295

Range calculation for float for decimal concept:

4294967296/2=214748365

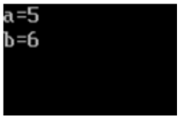


Therefore, the range will be (-214748365 to 214748364)

…… similarly rest data type range would be calculated by using graph.

Type	Size (bits)	Range
char or signed char	8	-128 to 127
unsigned char	8	0 to 255
int or signed int	16	-32768 to 32767
unsigned int	16	0 to 65535

OUTPUT:

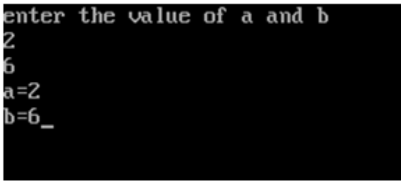


```
a=5
b=6
```

Program 3: Write a program to enter the value of two numbers and print these values.

```
#include<iostream.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
cout<<"enter the value of a and b";
scanf("%d%d",&a,&b);
cout<<"a="<<a<<"\n"<<"b="<<b;
getch();
}
```

OUTPUT:

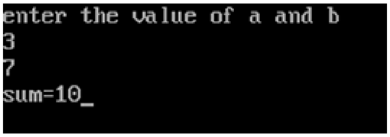


```
enter the value of a and b
2
6
a=2
b=6_
```

Program 4: Write a program to find the sum of two numbers.

```
#include<iostream.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
cout<<"enter the value of a and b";
cin>>a>>b;
c=a+b;
cout<<"sum="<<c;
getch();
}
```

OUTPUT:

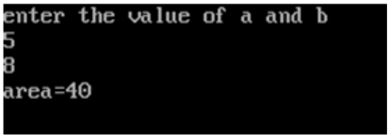


```
enter the value of a and b
3
7
sum=10_
```

Program 5: Write a program to find the area of rectangle.

```
#include<iostream.h>
#include<conio.h>
void main()
{
int a,b,area;
clrscr();
cout<<”enter the value of a and b”;
cin>>a>>b;
area=a*b;
cout<<”area=”<<area;
getch();
}
```

OUTPUT:



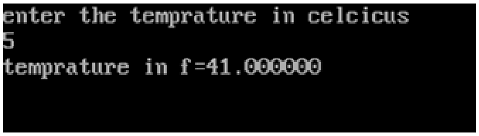
```
enter the value of a and b
5
8
area=40
```

Program 6: Write a program to find the simple interest.

```
#include<iostream.h>
#include<conio.h>
void main()
{
float p,r,t,si;
clrscr();
cout<<”enter the value of p,r and t”;
cin>>p>>r>>t;
si=(p*r*t)/100;
cout<<”simple interest=”<<si;
getch();
}
```

```
cin>>c;
f=(c/5)*9+32;
cout<<”temprature in Fahrenheit f=”<<f;
getch();
}
```

OUTPUT:

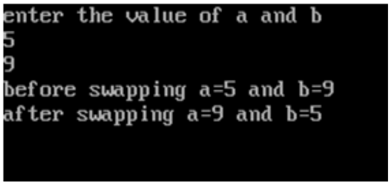


```
enter the temprature in celcicus
5
temprature in f=41.000000
```

Program 9: Write a program to swap two numbers using third variable

```
#include<iostream.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
cout<<”enter the value of a and b”;
cin>>a>>b;
cout<<”before swapping a=”<<a<<”and b=”<<b<<”\n”;
 c=a;
 a=b;
 b=c;
cout<<”after swapping a=”<<a<<” and b=”<<b;
getch();
}
```

OUTPUT:

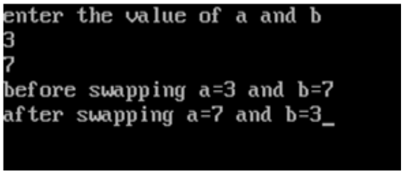


```
enter the value of a and b
5
9
before swapping a=5 and b=9
after swapping a=9 and b=5
```

Program 10: Write a program to swap two numbers without using third variable

```
#include<iostream.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
cout<<”enter the value of a and b”;
cin>>a>>b;
cout<<”before swapping a=”<<a<<” and b=”<<b<<”\n”;
a=a+b;
b=a-b;
a=a-b;
cout<<”after swapping a=”<<a<<‟ and b=”<<b;
getch();
}
```

OUTPUT:



```
enter the value of a and b
3
7
before swapping a=3 and b=7
after swapping a=7 and b=3_
```

❒ Key Points to Remember

1. Language of computer is used to control the behavior/activity of machine.
2. There are three types of computer language
   a. High level language
   b. Middle level language
   c. Low level language
3. Machine language is directly used by the motherboard/microprocessor.
4. Translator is used for converting/changing the high level language or low level language into machine level language.
5. The main function of every C++ program is main(), from where execution of a C++ program begins. The body of C++ program contains declaration, arithmetic and logical statements, control statements and function calls.

QUESTIONS AND ANSWERS

Objective Questions

1. Choose the correct output of the code.

```
#include<iostream.h>
void main()
{
char c = 's';
cout<<"c is "<<c<<"\n";
}
```

```
(a) c is s          (b) c is c
(c) s is c          (d) None of these
```

Answer: (a)

2. Choose the correct output of the code.

```
#include<iostream.h>
void main()
{
float f = 0.1;
if(f == 0.1)
cout<<"True";
else
cout<<"False";
}
```

```
(a) True            (b) False
```

Answer: (a)

3. Is there any invalid variable name declaration?

```
(a)  int _r3         (b)  int _3r
(c)  int _r3         (d)  all of above
```

Answer: (d)

4. Is there any invalid variable name declaration?

```
(a)  inr r3          (b)  int r_3
(c)  int _3r         (d)  int 3_r
```

Answer: (c)

5. Program without main() function is possible

(a) true (b) false

Answer: (b)

12. The 2 bytes are used for _____ data type.

(a) char

(b) double

(c) float

(d) int

Answer: (d)

13. What will be the range of int data type?

(a) -32768 to 32767

(b) -65536 to 65535

(c) 3.4e^34 to 3.4 e^(-34)

(d) None of above

Answer: (a)

14. Find the primitive data type.

(a) array

(b) structure

(c) pointer

(d) int

Answer: (d)

15. Size of integer data type is

(a) 4 bytes

(b) 8 bytes

(c) Cannot be determined

(d) Depends on the system/compiler

Answer: (d)

16. Choose the correct output

```
#include <iostream.h>
 int main()
 {
   char c;
   c = -128;
   cout<<c<<”\n”;
   return 0;
 }
```

(a) -128

(b) 128

(c) Depends on the compiler

(d) None of these

Answer: (a)

17. What is range of short int in C++ programming?

(a) -128 to 127

(b) -256 to 255

(c) -32768 to 32767

(d) All of the mentioned

Answer: (a)

18. Find the output of the following code:

```
#include <iostream.h>
 int main()
   {
    cout<<”C++ programming “<< “by satya”;
   }
```

(a) C++ programming by satya (b) by satya C++ programming

(c) both a and b (d) Compilation error

Answer: (a)

19. Find the output of the following code:

```
#include <iostream.h>
 #define A 10
 void main()
 {
   int A = 5;
   cout<<”A =”<< A;
 }
```

(a) A= 5 (b) A = 10

(c) Runtime error (d) Compilation error

Answer: (a)

20. Find the output of the following code:

```
#include <iostream.h>
  void main()
  {
    int a = 010;
    cout<< a;
  }
```

(a) 9 (b) 10

(c) 2 (d) 8

Answer: (b)

21. Find the output of the following code:

```
#include <iostream.h>
void main()
{
 cout<<”Hello dear!<<x<<”\n”;
}
```

(a) Hello dear! X;

(b) Hello dear! Followed by a junk value

(c) Hello dear!

(d) Compile time error

Answer: (d)

22. Find the output of the following code:

```
#include <iostream.h>
 void main()
 {
   int x = 10000;
   int x = 34;
   cout<<”Hello satya!”<<x<<”\n”;
 }
```

(a) Hello satya! 34

(b) Hello satya!1000

(c) Hello satya! Followed by a junk value

(d) Compile time error

Answer: (a)

23. Find out the invalid variable name declaration

(a) float iec = 3.14

(b) double iec = 3.14

(c) int iec = 3.14

(d) #define iec 3.14

Answer: (c)

24. Choose the correct alternative for the following code:

```
#include <iostream.h>
void main()
{
  int main = 3;
  cout<<main;
}
```

(a) Compile time error

(b) run time error

(c) main can not be used as a variable

(d) 3

Answer: (d)

Subjective Questions

1. What does main function do?
2. Explain and differentiate between cout and cin.
3. What is main header file of C++ language?
4. For what iostream stands? Differentiate between iostream.h and stdio.h.
5. Write steps to execute the programs.
6. Write the format of simple C++ program.
7. Define the role of cout function. Explain it by suitable example.
8. Define the getch() with suitable example.
9. Define the clrscr() with suitable example.
10. Why do we use programming? How does it help in our daily life?

Programming Practices

1. Write a program to add three numbers.
2. Write a program to swap two numbers.
3. Write a program to find the area of a triangle.
4. Write a program to calculate the average of four numbers.
5. Write a program to find c=(f-32)/5.
6. Write a program to calculate the area=l*b.
7. Write a program to find the area=3.14*r*r.
8. Write a program to calculate the D=b*b-4*a*c.
9. Write a program to calculate the area of a square.
10. Write a program to find the difference of two numbers.
11. Write a program to find f=a+b+c+d/e.
12. Write a program to find avg=(a+b+c+d+e+f)/6.
13. Write a program to find the sum of five numbers, that is, f=a+b+c+d+e.
14. Write a program to find the peri=2*(l+b).
15. Write a program to find the area=l*b.
16. Write a program to find the average of six numbers, that is, avg=(a+b+c+d+e+f)/6
17. Write a program to calculate a=0.5*l*b*h.
18. Write a program to calculate p=a+b+c*f-g/h.