INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

EEC-101
Programming with C++

Module-2.2 Constants, Variables, and Data Types

Ramanuja Panigrahi

Topics

- Concepts of algorithm & flow charts;
- Input/output,
- constants, variables
- operators;
- Naming conventions and styles;
- Conditions and selection statements;
- Looping and control structures (while, for, do-while, break and continue);
- File I/O, Header files, String processing;
- Pre-processor directives such as #include, #define, #ifdef, #ifndef;
- Compiling and linking.

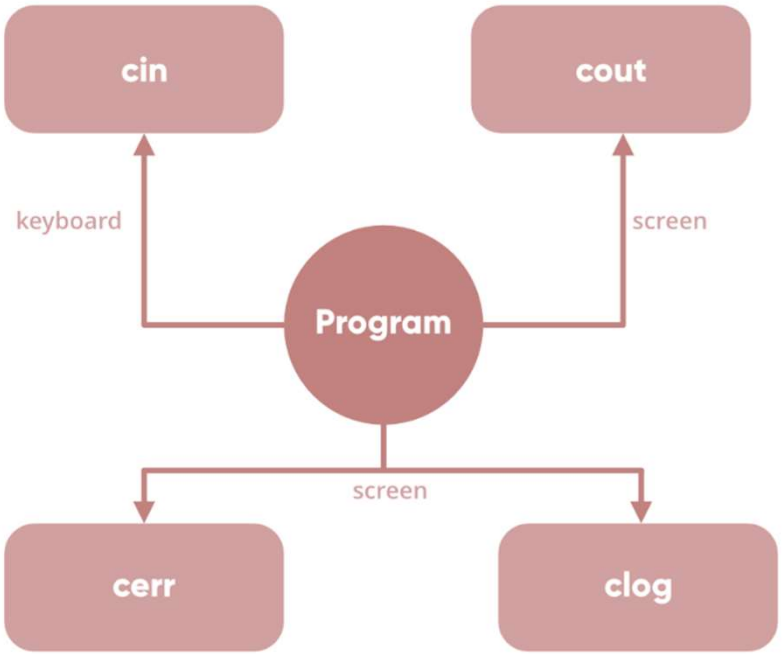
```cpp
/*******************************
Name: Addition of two numbers
Written by: Ramanuja Panigrahi
Date: 23rd August 2024
IIT Roorkee
*******************************/
// this program asks the user to enter
// two numbers and displays their sum as output
#include<iostream>
using namespace std;
int main ()
    {
     cout<<"This program will add two numbers"<<endl;
     cout<<"Enter First number"<<endl;
     int first_number{0};
     cin>>first_number;
     cout<<"Enter Second number"<<endl;
     int second_number{0};
     cin>>second_number;
     int sum;
     sum=first_number+second_number;
     cout<<"sum of "<<first_number<<" and "<<second_number<<" = "<<sum;
     return (0);

    }

```

Input/ output

- C++ comes with libraries that provide us with many ways for performing input and output. In C++ input and output are performed in the form of a sequence of bytes or more commonly known as **streams**.
- Input Stream: If the direction of flow of bytes is from the device(for example, Keyboard) to the main memory then this process is called input.
- Output Stream: If the direction of flow of bytes is opposite, i.e. from main memory to device(display screen) then this process is called output

Header files are available in C++ for Input/Output operations :

iostream: iostream stands for standard input-output stream. This header file contains definitions of objects like cin, cout, cerr, etc.



- In C++ after the header files, we often use '*using namespace std;*'.
- The reason behind it is that all of the standard library definitions are inside the namespace std. Therefore, in order to use them, we use *namespace std*.
- Otherwise, we need to write STD:: at every line (eg. STD::cout).

- The two instances **cout in C++** and **cin in C++** of iostream class are used very often for printing outputs and taking inputs, respectively.

- These two are the most basic methods of taking input and printing output in C++. To use cin and cout in C++ one must include the header file *iostream* in the program.

```
7   // this program asks the user to enter
8   // two numbers and displays their sum as output
9   #include<iostream>
10  using namespace std;
11  int main ()
12      {
13          cout<<"This program will add two numbers"<<endl;
14          cout<<"Enter First number"<<endl;
15          int first_number{0};
16          cin>>first_number;
```

Standard output stream (cout)

- Usually the standard output device is the display screen.
- The C++ **cout** statement is the instance of the ostream class.
- It is used to produce output on the standard output device which is usually the display screen.
- The data needed to be displayed on the screen is inserted in the standard output stream (cout) using the insertion operator(**<<**).

cout and <<

- cout is a pre-defined object in C++ that denotes the output stream.
- << is called insertion operator (binary).
- The operator does two things:
  - converts rhs into a sequence of characters,
  - inserts them into the output stream.

- Insert the data into the cout stream

```
cout << data;
```

- Can be chained

```
cout << "data 1 is " << data1;
```

- Does not automatically add line breaks

```
cout << "data 1 is " << data1 << endl;
cout << "data 1 is " << data1 << "\n";
```

```cpp
#include <iostream>

using namespace std;

int main()
{
    string sample {"EEC-101"};

    cout << sample << " is a 1st year B.Tech course"<<endl;

    return 0;
}
```

```
EEC-101 is a 1st year B.Tech course

Process returned 0 (0x0)   execution time : 0.005 s
Press any key to continue.
```

In the above program, the insertion operator(**<<**) inserts the value of the string variable **sample** followed by the string " is a 1st year B.Tech course“ in the standard output stream **cout** which is then displayed on the screen.

standard input stream (cin) and >>

- Usually the input device in a computer is the keyboard.
- In C++ cin statement is the instance of the class istream and is used to read input from the standard input device which is usually a keyboard.
- The extraction operator(>>) is used along with the object cin for reading inputs.
- The extraction operator extracts the data from the object cin which is entered using the keyboard.

standard input stream (cin) and >>

- Right side – only variable name(s)
- The operator does two things:
  - extracts necessary number of bytes from the input stream. (stops at the character that is not proper for the data type being extracted), (skips initial white spaces)
  - converts the extracted bytes into appropriate format and stores at the memory referred on rhs.
- Can be used several times in the same statement.

- Extract data from the cin stream based on data's type

```
cin >> data;
```

- Can be chained

```
cin >> data1 >> data2;
```

- Can fail if the entered data cannot be interpreted
  `data` could have an undetermined value

```cpp
#include <iostream>
using namespace std;

int main()
{
    int age;

    cout << "Enter your age:"<<endl;
    cin >> age;
    cout << "Your age is: " << age;

    return 0;
}
```

```
Enter your age:
21
Your age is: 21
Process returned 0 (0x0)   execution time : 5.473 s
Press any key to continue.
```

- The above program asks the user to input the age.
- The object cin is connected to the input device.
- The age entered by the user is extracted from cin using the extraction operator(**>>**) and the extracted data is then stored in the variable **age** present on the right side of the extraction operator.

```cpp
#include <iostream>
using namespace std;

int main()
{
    int i;
    char c;
    float f;

    cout << "Enter the input"<<endl;
    cin >> i>>c>>f;
    cout << "i="<<i<<endl;
    cout << "c="<<c<<endl;
    cout << "f="<<f<<endl;

    return 0;
}
```

```
Enter the input
20r10

i=20
c=r
f=10
```

```
Enter the input
20
h
10

i=20
c=h
f=10
```

```
Enter the input
20.405

i=20
c=.
f=405
```

```
Enter the input
20
hello

i=20
c=h
f=0
```

- Standard Input

```
#include <iostream>

cin >> var1 >> var2 >> var3;
```

- Whitespace used as delimiters
  - blanks, tabs, newlines
- Values must be compatible with data type of objects

```
cout << "Output generated"; // prints Output generated on screen
cout << 20; // prints number 20 on screen
cout << x; // prints the content of x on screen

cout << "Hello"; // prints Hello
cout << Hello; // prints the content of Hello variable
```

```
cout << "First sentence”<<‘\n ‘;
   cout << "Second sentence.\nThird sentence.";
```

This produces the following output:

```
First sentence.
Second sentence.
Third sentence.
```

```
cout << "First sentence." << endl;
   cout << "Second sentence." << endl;
```

- Would print out:
  First sentence.
  Second sentence.

- *int* age;

  cin >> age;
- The first statement declares a variable of type int called age, and the second one waits for an input from cin (the keyboard) in order to store it in this integer variable.
- cin can only process the input from the keyboard once the ENTER key has been pressed.
- Therefore, even if you request a single character, the extraction from cin will not process the input until the user presses ENTER after the character has been introduced.

  You must always consider the type of the variable that you are using as a container with cin extractions.
- If you request an integer you will get an integer, if you request a character you will get a character and if you request a string of characters you will get a string of characters.