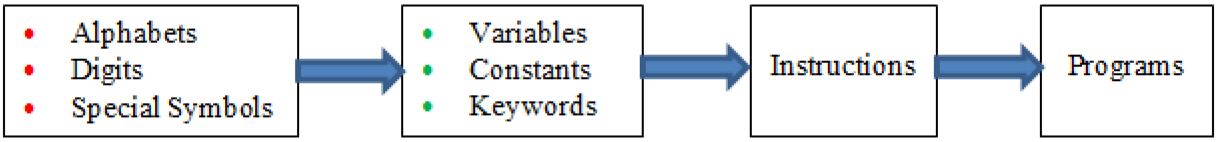
STEPS IN LEARNING C++



- Alphabets
- Digits
- Special Symbols

- Variables
- Constants
- Keywords

Instructions

Programs

One must understand character set used while writing C++ programs. Characters in C++ are grouped into the following categories:

1. **Letters or Alphabets**
   A, B, C,……………………..Z
   a, b, c, ………………………..z

2. **Digits**
   0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Keyboard

3. **Special symbols or characters**
   + - * # $ @ ? . , { ([})] etc

4. **White spaces**: also known as **blanks** or **blank spaces**

Note: A character set denotes any English letter (A, B, C, ……Z or a,b,c……..z), digits (0,1,2…..9) or special symbols (+, -, *, /, #, $ etc) used to represent information.

VARIABLES

Variables are used to store values. We can also define variables as identifiers i.e., names used to hold values.

RULES FOR NAMING VARIABLES

1. First character must be an alphabet [English Letter] or underscore symbol (_).
2. Variable names must not contain white spaces [BLANKS]
3. First character may be followed by a sequence of letters and digits
4. Keywords (also known as reserved words) must not be used as variable names. There are 63 keywords in C++ language – int, float, char, for, while, do const, struct, union, class, template etc.
5. No special characters except underscore is used in variable names.
6. In C++, uppercase and lowercase letters are different. Therefore the variable name SUM, Sum, sum are different variables. Good practice is to represent variable names in LOWER-CASE.

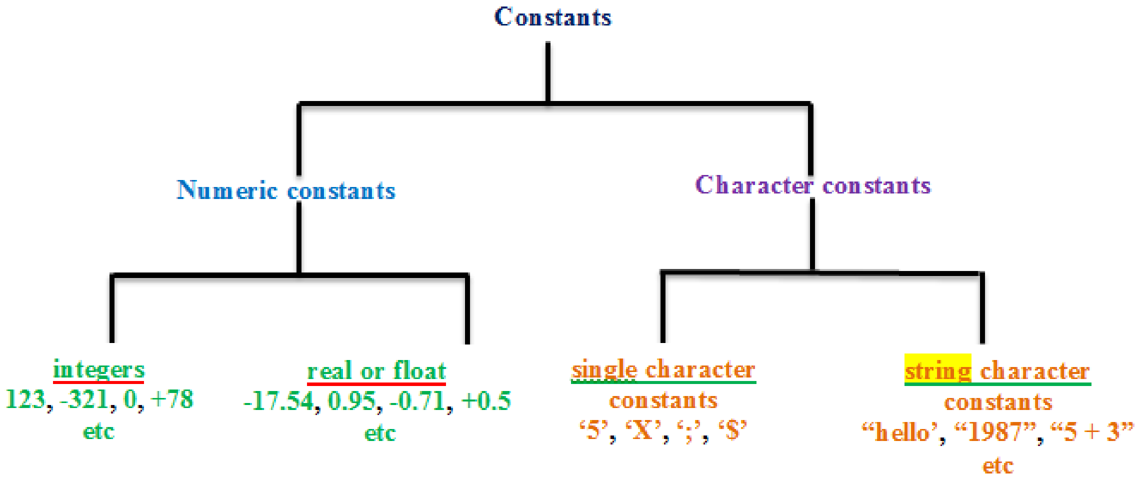
Note: It is useful to give meaningful names to the variables.

Valid names
Average
height
total
net_pay
_gopal
John
Delhi
a
b

Invalid names
5bc
int
float
gopal$
gopal 123
John
Delhi#

CONSTANTS

Constants in *C++* refer to fixed values. They do not change during execution of a program.

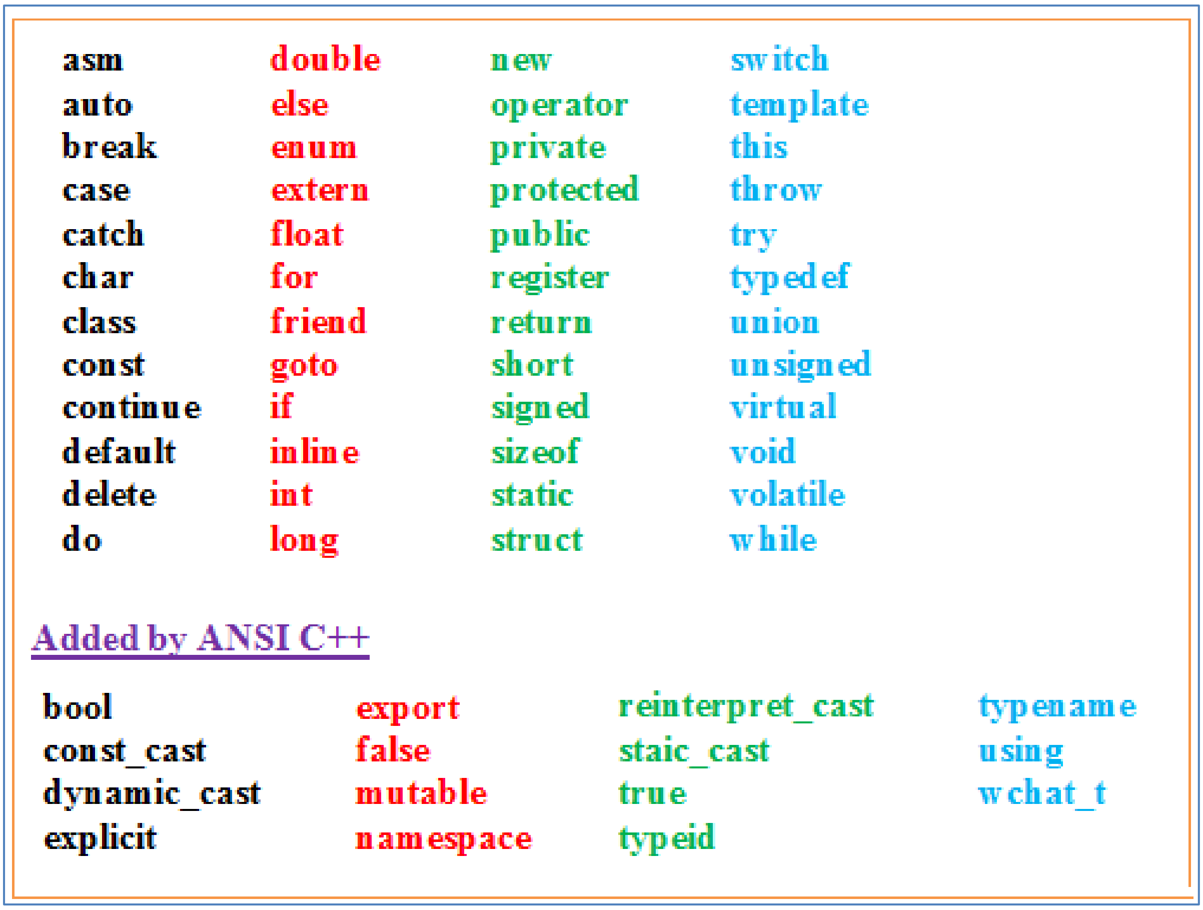


- Characters must be enclosed in single quotes
- Strings must be enclosed in double quotes

- Single character constants are enclosed in single quotes.
- String character constants are enclosed in double quotes.
- String = collection of characters
- Integers do not contain fractions.
- Real constants may contain fraction values.

KEYWORDS (Reserved words)

There are **63** keywords available in C++. Reserved/Key words have a special meaning within a C++ program. These symbols, the reserved words, must not be used for any other purposes. All reserved words are in lower-case letters. The following are reserved words of C++.



asm
auto
break
case
catch
char
class
const
continue
default
delete
do

double
else
enum
extern
float
for
friend
goto
if
inline
int
long

new
operator
private
protected
public
register
return
short
signed
sizeof
static
struct

switch
template
this
throw
try
typedef
union
unsigned
virtual
void
volatile
while

Added by ANSI C++

bool
const_cast
dynamic_cast
explicit

export
false
mutable
namespace

reinterpret_cast
staic_cast
true
typeid

typename
using
wchat_t

N.B. keywords have special meaning and purpose

cin AND *cout* OBJECTS

cin

Keyboard

cin is an input operator in C++ and pronounced as "C in". Note that cin uses >> *extraction operator* or *get from operator*.

For example, cin>>*x*; extracts (or takes) the value from the keyboard and assigns it to the variable *x* on its right. This corresponds to the familiar scanf() operation in C language.

The statement cin>>x; is an input statement and causes the program to wait for the user to type a number (if x is defined as int or float).

Note that cin is a predefined object in C++ that corresponds to the standard input stream, which represents the keyboard. cin is used to supply input to the program from standard input device KEYBOARD. cin receives values from keyboard. Format of cin object is as follows:

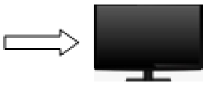
```
cin>>variable1>>variable2>> ….;
```

```
cin>>a>>b>>c;
```

here a, b, c are variable names of type int.

Note: **cin** is counterpart of cout.

Cout



cout is output operator in C++ and pronounced as "C out". Note that cout uses << *insertion operator* or *put to operator*.

For example, cout<<*x*; sends the value in variable *x* to SCREEN. This corresponds to the familiar printf() operation in C language.

Note that cout is a predefined object in C++ that corresponds to the standard output stream, which represents the screen or printer.
General form of cout object is as follows:

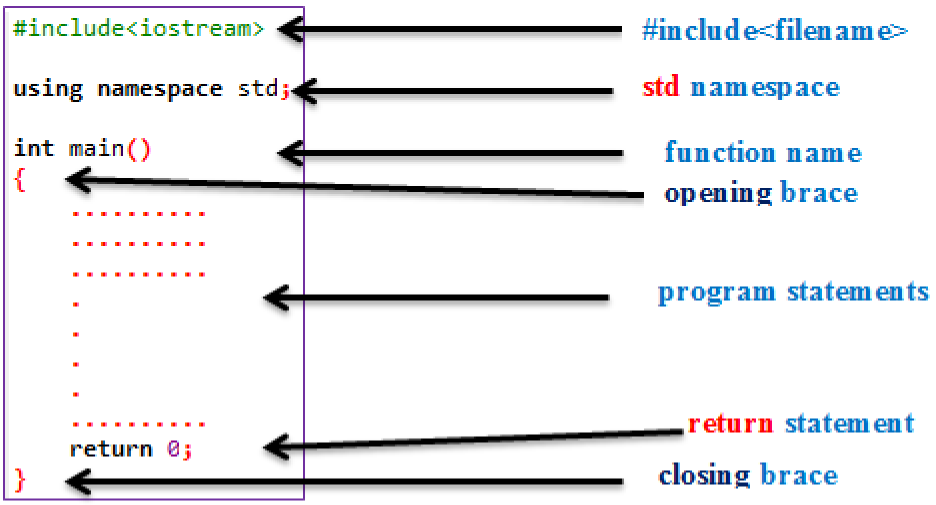
```
cout<<variable1<<variable2<<….;
```

cout has two forms:

1. Simply sends text in double quotes to screen
   cout<<"GOPALAAAAAAAA";
2. Sends values in the program to screen
   cout<<a<<b<<c;

   here values of variables a,b,c are printed on screen.

FORMAT OF SIMPLE C++ PROGRAM



```
#include<iostream>

using namespace std;

int main()
{
    ..........
    ..........
    ..........
    .
    .
    .
    .
    ..........
    return 0;
}
```

Note

#include is a pre-processor directive and is written as #include<filename>. Filename is the name of the library file that contains the required object definition.

Consider the below statement:

#include<iostream>

This directive causes the pre-processor to add the contents of the iostream file to the program. It contains declarations for the identifier cin or cout and the operator >> or <<.

The header file iostream should be included at the beginning of all programs that use input/output statements.

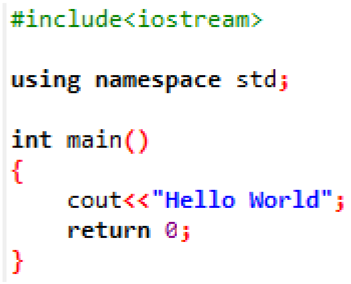
using namespace std;

Namespace is the new concept introduced by the ANSI C++ standards committee. C++ standard class libraries are defined in std namespace. All ANSI C++ programs must include this directive. **using** and **namespace** are the new keywords of C++. std stands for standard. This includes objects like cout, cin, string, vector, map, etc. Because these objects are used so commonly, it's popular to add "using namespace std" at the top of program. Note that namespace makes our task easy, but not necessary.

In C++, main() returns an integer type value. Therefore every main() in C++ should end with a return 0 statement. Note that the default return type for all functions in C++ is int.

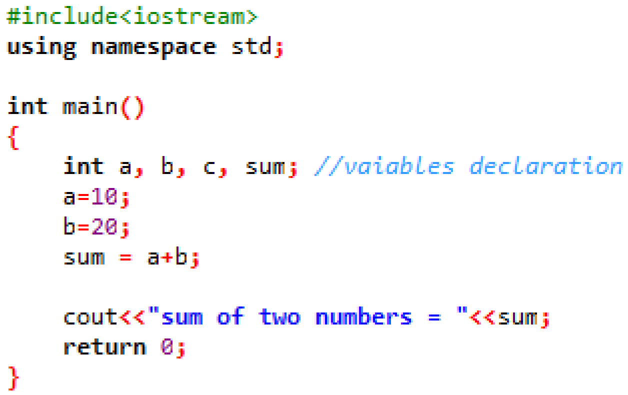
SIMPLE PROGRAMS

Program 1: Hello World



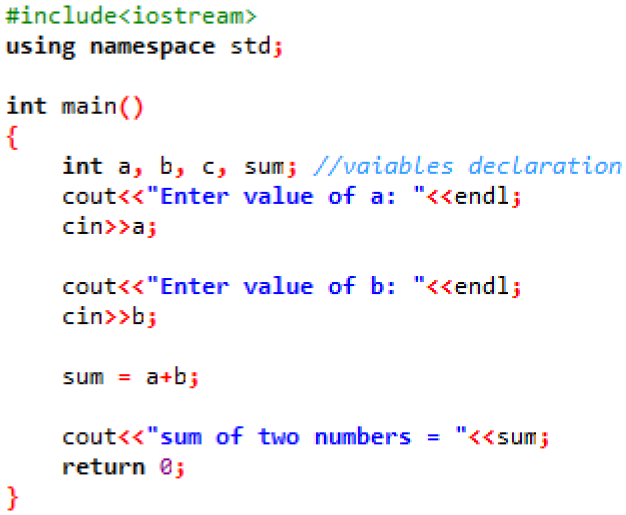
```
#include<iostream>

using namespace std;

int main()
{
    cout<<"Hello World";
    return 0;
}
```

Program 2: Sum of given two numbers
(inputs are directly coded in the program)



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

int main()
{
    int a, b, c, sum; //vaiables declaration
    a=10;
    b=20;
    sum = a+b;

    cout<<"sum of two numbers = "<<sum;
    return 0;
}
```

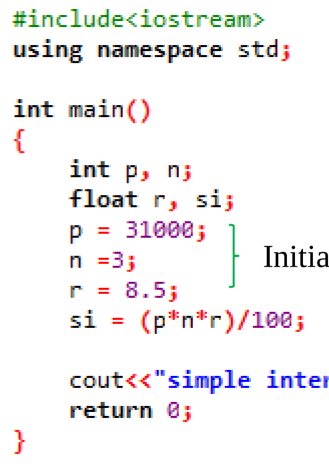
Sum of given two numbers
(using cin)



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

int main()
{
    int a, b, c, sum; //vaiables declaration
    cout<<"Enter value of a: "<<endl;
    cin>>a;

    cout<<"Enter value of b: "<<endl;
    cin>>b;

    sum = a+b;

    cout<<"sum of two numbers = "<<sum;
    return 0;
}
```

- Count the number of variables required in the program (in this program 3 variables required)
- Above program requires 2 input variables (a, b) and 1 output variable sum
- Declare type of variables. Here a, b and sum are declared as int type
- Decide how to give input to the program – either direct initialization or using cin

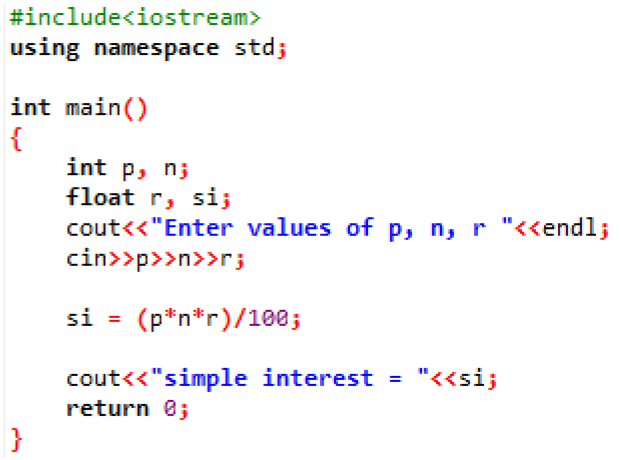
Program 3: Calculation of simple interest
(inputs are directly coded in the program)



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

int main()
{
    int p, n;
    float r, si;
    p = 31000;
    n =3;
    r = 8.5;
    si = (p*n*r)/100;

    cout<<"simple interest = "<<si;
    return 0;
}
```

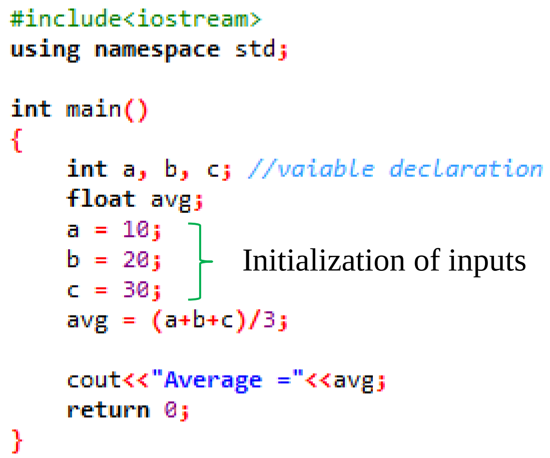
Initialization of inputs

Calculation of simple interest
(using cin)



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

int main()
{
    int p, n;
    float r, si;
    cout<<"Enter values of p, n, r "<<endl;
    cin>>p>>n>>r;

    si = (p*n*r)/100;

    cout<<"simple interest = "<<si;
    return 0;
}
```

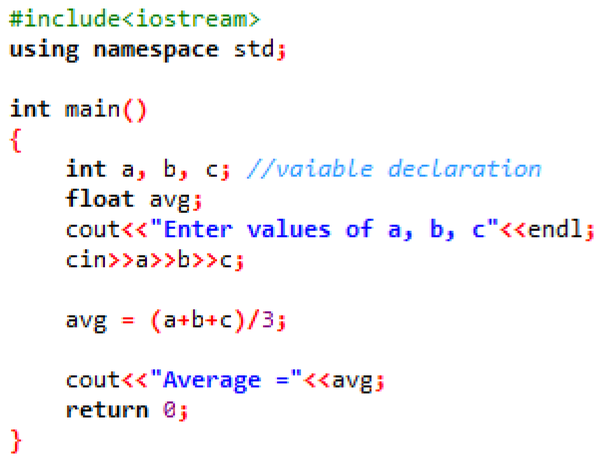
Program 4: calculation of average of 3 numbers
(inputs are directly coded in the program)



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

int main()
{
    int a, b, c; //vaiable declaration
    float avg;
    a = 10;
    b = 20;
    c = 30;
    avg = (a+b+c)/3;

    cout<<"Average ="<<avg;
    return 0;
}
```

Initialization of inputs

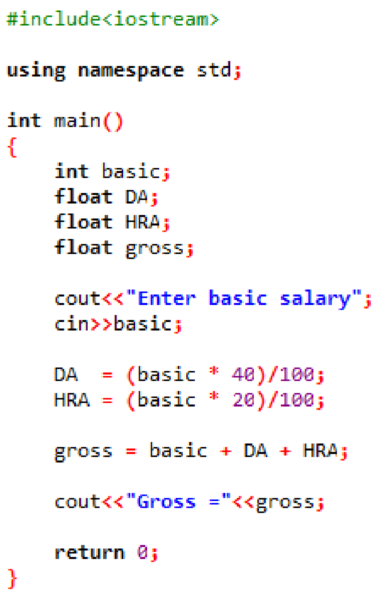
calculation of average of 3 numbers
(using cin)



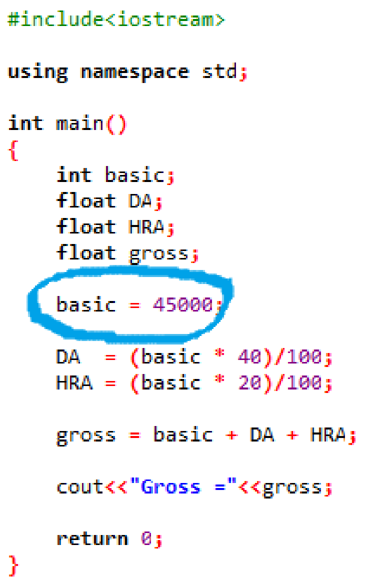
```
#include<iostream>
using namespace std;

int main()
{
    int a, b, c; //vaiable declaration
    float avg;
    cout<<"Enter values of a, b, c"<<endl;
    cin>>a>>b>>c;

    avg = (a+b+c)/3;

    cout<<"Average ="<<avg;
    return 0;
}
```

Program 5

Suresh basic salary is input through the keyboard. His DA is 40% of basic salary and HRA is 20% of basic salary. Write a C++ program to calculate his gross salary.



```
#include<iostream>

using namespace std;

int main()
{
    int basic;
    float DA;
    float HRA;
    float gross;

    cout<<"Enter basic salary";
    cin>>basic;

    DA  = (basic * 40)/100;
    HRA = (basic * 20)/100;

    gross = basic + DA + HRA;

    cout<<"Gross ="<<gross;

    return 0;
}
```

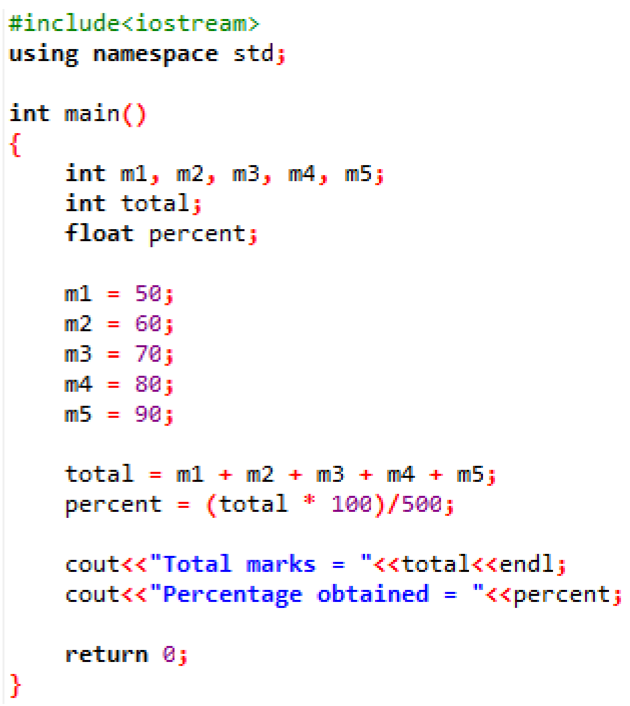


```
#include<iostream>

using namespace std;

int main()
{
    int basic;
    float DA;
    float HRA;
    float gross;

    basic = 45000;

    DA  = (basic * 40)/100;
    HRA = (basic * 20)/100;

    gross = basic + DA + HRA;

    cout<<"Gross ="<<gross;

    return 0;
}
```

Program 6

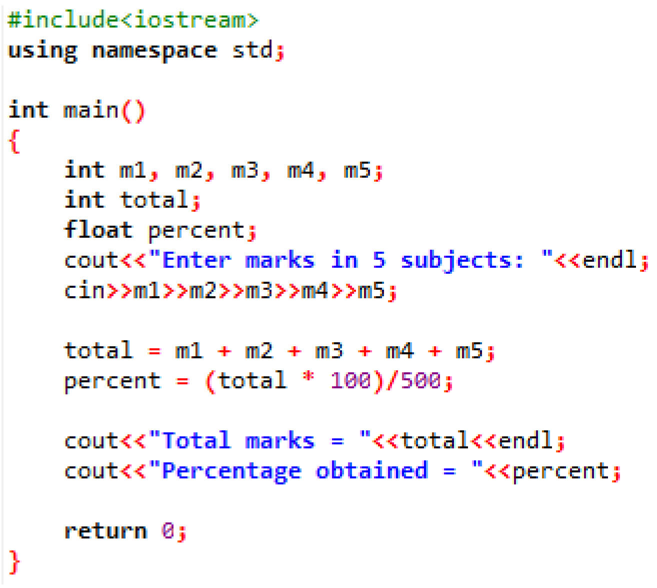
If the marks obtained by a student in 5 different subjects are input through the keyboard, find the aggregate marks and % of marks obtained by the student. Assume that the maximum marks that can be obtained by a student in each subject is 100.

Program 5: Calculation of percentage
(inputs are directly coded in the program)



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

int main()
{
    int m1, m2, m3, m4, m5;
    int total;
    float percent;

    m1 = 50;
    m2 = 60;
    m3 = 70;
    m4 = 80;
    m5 = 90;

    total = m1 + m2 + m3 + m4 + m5;
    percent = (total * 100)/500;

    cout<<"Total marks = "<<total<<endl;
    cout<<"Percentage obtained = "<<percent;

    return 0;
}
```

Calculation of percentage
(using cin)



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

int main()
{
    int m1, m2, m3, m4, m5;
    int total;
    float percent;
    cout<<"Enter marks in 5 subjects: "<<endl;
    cin>>m1>>m2>>m3>>m4>>m5;

    total = m1 + m2 + m3 + m4 + m5;
    percent = (total * 100)/500;

    cout<<"Total marks = "<<total<<endl;
    cout<<"Percentage obtained = "<<percent;

    return 0;
}
```

EXERCISES

1. Using arithmetic operators in C++, write a program that declares and assigns values to the variables ***a***, ***b***, ***c*** and then does the following:

    a. Halves the value of ***a***
    b. Doubles ***b***
    c. Multiplies ***c*** by itself
    d. Prints out the results of above operations

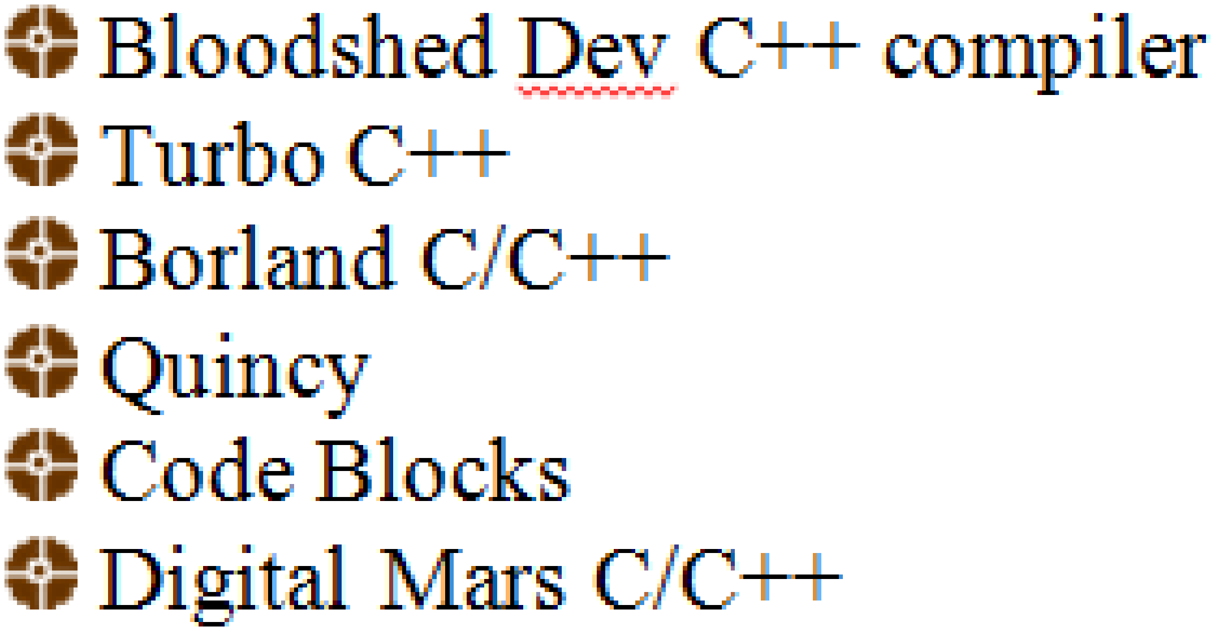
2. Enter the temperature in Celsius and convert that into Fahrenheit. Formula: $F = \frac{9}{5} * C + 32;$

3. Two variables ***a*** and ***b*** contain values of 10 and 20. Write a program to interchange the contents of ***a*** and ***b***.

4. Write a program to find area of a triangle. Input the length and breadth of a rectangle through the keyboard. Formula: Area = length * breadth

5. Write a program to find area of a circle of any radius (r). Formula: $Area = \pi r^2$

C/C++ compilers



- Bloodshed Dev C++ compiler
- Turbo C++
- Borland C/C++
- Quincy
- Code Blocks
- Digital Mars C/C++