

## Lecture 4

### Not:

- The variable list may consist of one or more identifier names separated by commas. Some valid declarations are shown here:

#### Example:

```
int    i, j, k;
char   c, ch;
float  f, salary;
double d;
```

The line `int i, j, k;` both declares and defines the variables `i`, `j` and `k`; which instructs the compiler to create variables named `i`, `j` and `k` of type `int`.

- Variables can be initialized (assigned an initial value) in their declaration. The initializer consists of an equal sign followed by a constant expression as follows:

#### Example:

```
int    d = 3, f = 5; // definition and initializing d and
f.
float  A1 = 2.2;    // definition and initializes 2.
char   x = 'x';    // the variable x has the value 'x'.
```

### 4.1 C++ Program Structure

Let us look at a simple code that would print the words Hello World.

```
#include <iostream>

using namespace std;

int main() // main() is where program execution begins.
{
    cout << "Hello World"; // prints Hello World
    return 0;
}
```

### Let us look at the various parts of the above program:

1. The C++ language defines several headers, which contain information that is either necessary or useful to your program. For this program, the header `<iostream.h>` is needed for output string in the screen.
2. `int main()` : is the main function where program execution begins.
3. `//` : is a single-line comment available in C++. Single-line comments begin with `//` and stop at the end of the line.
4. `cout << " : This is my first C++ program."`; causes the message "This is my first C++ program" to be displayed on the screen.
5. `<<` : it is the send operator
6. `return 0`: terminates `main()` function and causes it to return the value 0 to the calling process.
7. `;` : semicolon , its used as terminator for every C++ statement.

❖ The **OUTPUT** for this program is :



**Hello World**

## 4.2 Standard Output (cout)

**cout**: the standard output of a program is the screen, and the C++ stream object defined to access it is `cout`. The `<` **operator** is overloaded to output data items of built-in types integer, float, double, strings and pointer values.

### Example:

```
cout << "Output sentence"; // prints Output sentence on screen  
cout << 120; // prints number 120 on screen
```

```
cout << x;                // prints the content of x on screen
```

### 4.3 Standard input (cin)

**cin**: is the input stream object, its read the input value from keyboard.

**>>** : it is the operator use to get from operator.

**endl** : is used to add a new-line at the end of the line.

- You can also use cin to request more than one datum input from the user:

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

is equivalent to:

```
cin >> a;
```

```
cin >> b;
```

- In both cases the user must give two data, one for variable a and another one for variable b that may be separated by any valid blank separator: a space, a tab character or a newline.

#### Example:

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

int main()
{
    char name;
    cout << "Please enter your name: ";
    cin >> name;
    cout << "Your name is: " << name << endl;
    return 0;
}
```

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

int main()
{
    cout << "This is a sentence,";
    cout << "This is another sentence.";
    return 0;
}
```

Following the output without any line break between them.

**This is a sentence,This is another sentence.**

### Example:

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

int main ()
{
    int i;
    cout << "Please enter an integer value: ";
    cin >> i;
    cout << "The value you entered is " << i;
    cout << " and its double is " << i*2 << endl;
    return 0;
}
```

❖ The **OUTPUT** for this program: will be shown on the screen:

**Please enter an integer value: 702**  
**The value you entered is 702 and its double is 1404.**

## Lecture Five

### C++ Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical calculations on operands(variables).

### Types of operators available in C++

- Arithmetic / Mathematical operator
- Assignment operator
- Increment Decrement operator
- Relational operator
- Logical operator
- Unary operator

### Arithmetic Operator:

There are following arithmetic operators supported by C++ language:

Assume variable A holds 10 and variable B holds 20, then:

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiplies both operands	A * B will give 200
/	Divides numerator by denominator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0

### **Increment Decrement operator**

Increment Decrement operators increase or decrease the operand by one value .

**Example: Assume A=10, find the output result for the following expressions:**

++	Increment operator, increases integer value by one	A++ will give 11
--	Decrement operator, decreases integer value by one	A-- will give 9

### **Assignment operator**

Assignment operator is used to copy value from right to left variable.

Suppose we have:

float X = 5, Y = 2;

<b>=</b>	Equal sign Copy value from right to left.	X = Y, Now both X and Y have 2
<b>+=</b>	<b>Plus Equal operator</b> to increase the left operand by right operand.	X+=5 → X=X+5 will give X= 10
<b>-=</b>	<b>Minus Equal operator</b> will return the subtraction of right operand from left operand.	Y-=1 → Y= Y-1 will give Y=1
<b>*=</b>	<b>Multiply Equal operator</b> will return the product of right operand and left operand.	X *= Y → X = X * Y, X = 10
<b>/=</b>	<b>Division Equal operator</b> will divide right operand by left operand and return the quotient.	X /= Y → X = X / Y, X = 2.5
<b>%=</b>	Modulus Equal to operator will divide right operand by left operand and return the mod ( Remainder ).	X %= Y is similar to X = X % Y, now X is 1

**Examples:**

**Rewrite the equivalent statements for the following expressions and find the results, assume X=2, Y=3, Z=4, V= 12, C=8.**

Example	Equivalent Statement	Result
X += 5	X = X + 5	X ← 7
Y -= 8	Y = Y - 8	Y ← -5
Z *= 5	Z = Z * 5	Z ←
V /= 4		V ←
C %= 3		C ←

**Relational Operator:**

Relational operators are used for checking conditions whether the given condition is true or false. If the condition is true, it will return non-zero value, if the condition is false, it will return 0.

Suppose we have,  
int X = 5, Y = 2;

Operator	Name	Description	Example
>	Greater than	Check whether the left operand is greater than right operand or not.	(X > Y) will return true
<	Smaller than	Check whether the left operand is smaller than right operand or not.	(X < Y) will return false
>=	Greater than or Equal to	Check whether the left operand is greater or equal to right operand or not.	(X >= Y) will return true
<=	Smaller than or Equal to	Check whether the left operand is smaller or equal to right operand or not.	(X <= Y) will return false
==	Equal to	Check whether the both operands are equal or not.	(X == Y) will return false
!=	Not Equal to	Check whether the both operands are equal or not.	(X != Y) will return true

Operator	Name	Example
==	Equality	5 == 5 // gives 1
!=	Inequality	5 != 5 // gives 0



<	<b>Less Than</b>	<b>5 &lt; 5.5 // gives 1</b>
<=	<b>Less Than or Equal</b>	<b>5 &lt;= 5 // gives 1</b>
>	<b>Greater Than</b>	<b>5 &gt; 5.5 // gives 0</b>
>=	<b>Greater Than or Equal</b>	<b>6.3 &gt;= 5 // gives 1</b>

### Logical Operators

Logical operators are used in situation when we have more than one condition in a single if statement.

Suppose we have,  
int X = 5, Y = 2;

Operator	Name	Description	Example
&&	AND	Return true if all conditions are true, return false if any of the condition is false.	if(X > Y && Y < X) will return true
	OR	Return false if all conditions are false, return true if any of the condition is true.	if(X > Y    X < Y) will return true
!	NOT	Return true if condition is false, return false if condition is true.	if(!(X > Y)) will return false

Operator	Name	Example
&&	Logical And	5 < 6 && 6 < 6 // gives 0
	Logical Or	5 < 6    6 < 5 // gives 1
!	Logical Negation (Not)	!(5 == 5) // gives 0

AND (&&) Table:		
A	B	A && B
T	T	T
T	F	F
F	T	F
F	F	F

AND (&&) Table:		
A	B	A && B
1	1	1
1	0	0
0	1	0
0	0	0

OR (  ) Table:		
A	B	A    B
T	T	T
T	F	T
F	T	T
F	F	F

OR (  ) Table:		
A	B	A    B
1	1	1
1	0	1
0	1	1
0	0	0

NOT (!) Table:	
A	!A
T	F
F	T

NOT (!) Table:	
A	!A
1	0
0	1

**Examples: The following example to understand all the arithmetic operators available in C++.**

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

main()
{
int a = 21;
int b = 10;
int c ;
c = a + b;
cout << "Line 1 - Value of c is :" << c << endl ;

c = a - b;
cout << "Line 2 - Value of c is :" << c << endl ;

c = a * b;
cout << "Line 3 - Value of c is :" << c << endl ;

c = a / b;
cout << "Line 4 - Value of c is :" << c << endl ;

c = a % b;
cout << "Line 5 - Value of c is :" << c << endl ;

c = a++;
cout << "Line 6 - Value of c is :" << c << endl ;

c = a--;
cout << "Line 7 - Value of c is :" << c << endl ;

return 0;
```

**The output for the above program is:**

```
Line 1 - Value of c is :31
Line 2 - Value of c is :11
Line 3 - Value of c is :210
Line 4 - Value of c is :2
Line 5 - Value of c is :1
Line 6 - Value of c is :21
Line 7 - Value of c is :22
```

Q/ What's Output:

```
#include<iostream>
using namespace std;
int main()
{ int x,y,z;
x=y=z=0;
x=++y + ++z;
cout<<x<<y<<z<<endl;
x=++y - --z;
cout<<x<<y<<z<<endl;
return 0;
}
```

Example: find the output result for the following logical operations:

Assume a=4, b=5, c=6

a=4, b=5, c=6

$(a < b) \&\& (b < c)$	$(a < b) \ \  (b > c)$	$!(a < b) \ \  (c > b)$	$(a < b) \ \  (b > c) \&\& (a > b) \ \  (a > c)$
T && T	T    T	!(T)    T	T    F && F    F
T	T	F    T	T    F    F
		T	T    F
			T

**Example: find the output result for the following logical operations:**

Assume:  $X=0, Y=1, Z=1$ . Find the following expression:

$M = ++X \ || \ ++Y \ \&\& \ ++Z$

$M = ++X \ || \ ++Y \ \&\& \ ++Z$

$= 1 \ || \ (2 \ \&\& \ 2)$

$= T \ || \ (T \ \&\& \ T)$

$= T \ || \ T$

$= T$

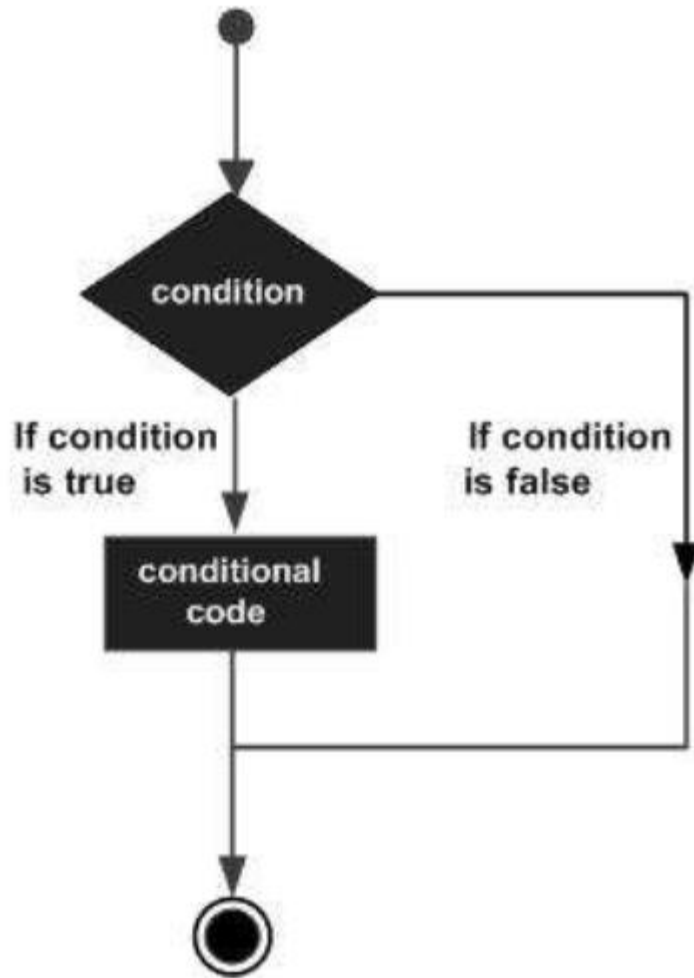
$= 1$

## Lecture 6

### **DECISION-MAKING STATEMENTS**

Decision making structures require that the programmer specify one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Following is the general form of a typical decision making structure found in most of the programming languages:



C++ programming language provides following types of decision making statements.

Statement	Description
if statement	An 'if' statement consists of a boolean expression followed by one or more statements.
if...else statement	An 'if' statement can be followed by an optional 'else' statement, which executes when the boolean expression is false.
switch statement	A 'switch' statement allows a variable to be tested

## **If Statement**

**if** statement consists of a boolean expression followed by one or more statements.

### **Syntax**

The syntax of an if statement in C++ is:

```
if(boolean_expression)
{
    // statement(s) will execute if the boolean expression is true
}
```

If the boolean expression evaluates to **true**, then the block of code inside the if statement will be executed. If boolean expression evaluates to **false**, then the first set of code after the end of the if statement (after the closing curly brace) will be executed.

### **Example:**

**Write C++ program to read a given integer value from keyboard and print the value if it is positive.**

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

int main()
{
int a;

cout << "Input integer value a :";

cin >>a;

if (a>0)
    cout<<"a is positive number" << endl;
```



The output for the above program is :

the input value

```
Input integer value a : 10
```

```
a is positive number
```

```
the value of a is:10
```

## if...else Statement

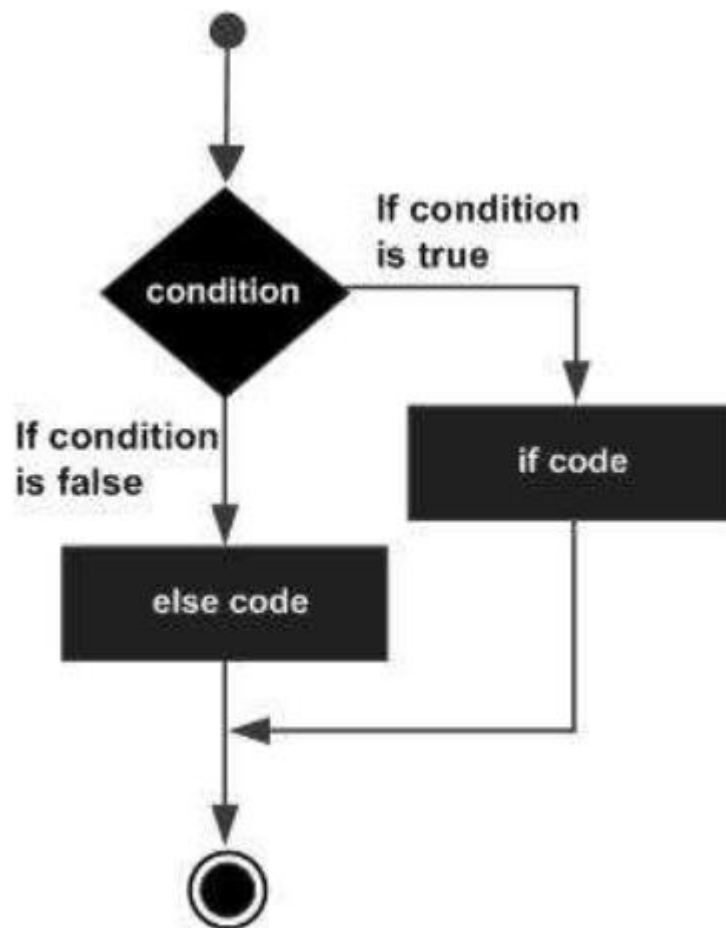
**if** statement can be followed by an optional **else** statement, which executes when the boolean expression is **false**.

## Syntax

The syntax of an if...else statement in C++ is:

```
if(boolean_expression)
{
    // statement(s) will execute if the boolean expression is true
}
else
{
    // statement(s) will execute if the boolean expression is false
}
```

If the boolean expression evaluates to **true**, then the **if block** of code will be executed, otherwise **else block** of code will be executed.



**Example:**

**Write C++ program to read a given integer value from keyboard and print the value if it is positive otherwise print it is negative**

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

int main()
{
int a;

cout << "Input integer value a :";

cin >>a;

if (a>0)
    cout<<"a is positive number" << a;

else
    cout <<"a is negative number"<< a;

return 0;
}
```

**Ex/Write C++ program to read a given integer value from keyboard and check if the value is even or odd .**

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

int main()
{
int a;
cout << "Input integer value a :";
cin >>a;
if (a % 2 == 0)
    cout<<"a is even number" << a;
else
    cout <<"a is odd number"<< a;
return 0;
}
```

### Example

Write C++ program to calculate Z value according to the following equations:

$$Z = \begin{cases} X + 10 & \text{if } X > 0 \\ 2X + 50 & \text{if } X < 0 \end{cases}$$

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

int main()
{
int X, Z;
cout << "Input integer value X :";

cin >>X;

if (X > 0)
{
Z=X+10;
cout<<" Z value is:" << Z;
}
else
{
Z= 2*X+50;
cout <<"Z value is :"<< Z;
}
return 0;
}
```

س/اكتب برنامج لقراءة رمز وتحديد هل هو من الاحرف الانكليزية الصغيرة او الكبيرة او رقم؟

**Q //write a program to enter a character and determine the character is digit,small or capital letter?**

```
#include <iostream>
using namespace std;
int main()
{ char letter;
  cout<<"Enter a letter:"; cin >> letter;
  if(letter >= '0' && letter <= '9') cout<<"entered a digit.";
  else
  if(letter >= 'a' && letter <= 'z') cout<<"entered a small letter.";
  else
  if(letter >= 'A' && letter <= 'Z') cout<<"entered a capital letter.";
  else
  cout<<"You entered a special letter.";
  return 0;
}
```

اكتب برنامج لمحاكاة عمل الحاسبة اليدوية

**Q//write program to simulation a simple hand Calculator work?**

```
#include<iostream>
using namespace std;
int main()
{
  int x,y; char sign;
  cin>>x >>sign >>y;
  if (sign== '+')
```

```

        cout<<x + y;
    else if (sign == '-')
        cout<< x - y;
    else if (sign == '*')
        cout<< x * y;
    else if (sign == '/')
        cout<< x / y;
    else
    cout << "ERROR" ;
    return 0;
}

```

س/اكتب برنامج لادخال رقم بين (1-7) ، وطباعة اسم اليوم المقابل من ايام الاسبوع؟

```

#include<iostream>
using namespace std;
int main()
{ int d; cout<<"Enter Number (1-7): "; cin>>d;
  if (d==1) cout<<"Sun.";
  else if (d==2) cout<<"Mun.";
  else if (d==3) cout<<"Tues.";
  else if (d==4) cout<<"Wen.";
  else if (d==5) cout<<"Thri.";
  else if (d==6) cout<<"Fri.";
  else if (d==7) cout<<"Sat.";
  else cout<<"number out of range!!!";
return 0;
}

```

## The Switch Statement :

The switch statement provides a way of choosing between a set of alternatives, based on the value of an expression. The general form of the switch statement is:

```
switch (expression) {  
  case constant 1: statements;  
  ...  
  ...  
  case constant n: statements;  
  default: statements;  
}
```

س/اكتب برنامج لادخال رقم بين (1-7) ، وطباعة اسم اليوم المقابل من ايام الاسبوع؟  
**Q // Write Program (W.P.) to read number between (1-7) and print name of day?**

```
#include<iostream>  
using namespace std;  
int main()  
{ int d;  
  cout<<"Enter Number (1-7): "; cin>>d;  
  switch (d)  
  {  
    case 1: cout<<"Sun."; break;  
    case 2: cout<<"Mun."; break;  
    case 3: cout<<"Tues."; break;  
    case 4: cout<<"Wen."; break;  
    case 5: cout<<"Thri."; break;  
    case 6: cout<<"Fri."; break;  
    case 7: cout<<"Sat."; break;  
    default: cout<<"number out of range!!!";  
  } return 0;  
}
```

اكتب برنامج لمحاكاة عمل الحاسبة اليدوية

**Q//write program to simulation a simple hand Calculator work?  
(using switch)**

```
#include<iostream>
using namespace std;
int main()
{ float a,b; char sign;
cin>>a>>sign>>b;
switch (sign)
{
case '+': cout<<"="<<a+b <<endl ; break;
case '-': cout<<"="<<a-b <<endl ; break;
case '*': cout<<"="<<a*b <<endl ; break;
case '/': cout<<"="<<a/b <<endl ; break;
default:cout<<"ERROR";
}
return 0;
}
```

**and we can use nested switch statement!!**

```
switch (expression1)
{
case constant 1: switch (expression2)
:
:
case constant n:

default: statements;
}
```

س/اكتب برنامج لقراءة اول حرفيين من بداية اليوم وطباعة اسم اليوم؟

**Q5//Write Program (W.P.) to read first and second letter from names day starting and print name of day?**

```
#include<iostream>
```



```

using namespace std;
int main()
{
char t1; char t2;
cout<<"Enter first charater: "; cin>>t1;
switch (t1)
{
case 's': {cout<<"Enter second charater: ";
cin>>t2;
switch(t2)
{ case 'u': cout<<"Sun."; break;
case 'a': cout<<"Sat."; break;
default: cout<<"ERROR !!!"; break;
}break; }
}
}

```

**H.W. : Complete the previous program.**

س/اكتب برنامج لقراءة 3 ارقام صحيحة وايجاد الاكبر بينهم؟

**Q7////write a program to input 3 integers and determine which of them is biggest? (Second Method)**

```

#include<iostream>
using namespace std;
int main()
{
int x,y,z; int larg;
cin>>x>>y>>z;
larg=x;
if (y>larg) larg=y;
if (z>larg) larg=z;
cout<<"larg No.="<<larg;
}

```

```
return 0;
}
```

```
//find smallest No.?  
#include<iostream>  
using namespace std;  
int main()  
{  
int x,y,z; int small;  
cin>>x>>y>>z;  
small=x;  
if (y<small) small=y;  
if (z<small) small=z;  
cout<<small;  
return 0;  
}
```

س/اكتب برنامج لقراءة اول حرفيين من بداية اليوم وطباعة اسم اليوم؟

**Q5//Write Program (W.P.) to read first and second letter from names day starting and print name of day?**

```
#include<iostream>  
using namespace std;  
int main()  
{  
char t1,t2;  
cout<<"Enter first charater: "; cin>>t1;  
switch (t1)  
{  
case 's': {cout<<"Enter second charater: ";  
cin>>t2;
```

```

switch(t2)
{ case 'u': cout<<"Sun."; break;
  case 'a': cout<<"Sat."; break;
  default: cout<<"ERROR !!!"; break;
}break; }
case 'm': cout<<"Mun."; break;
case 't': {cout<<"Enter second charater: ";
cin>>t2;
switch(t2)
{ case 'u': cout<<"Tues."; break;
  case 'h': cout<<"Thri."; break;
  default: cout<<"ERROR !!!";
} break; }
case 'w': cout<<"wen."; break;
case 'f': cout<<"Fri."; break;
default: cout<<"ERROR!!!";
}
return 0; }

```

س/اكتب برنامج لايجاد قيمة  $y$  من المعادلة التالية؟

**Q3// Write Program (W.P.) to find Y value , when**

$$Y = \sqrt{x^2 + z^2}$$

```

#include<iostream>
#include<math.h>
using namespace std;
int main()
{
int x,z; double y;
cout<<"Enter x value: "; cin>>x;

```

```
cout<<"Enter z value: "; cin>>z;
    y=sqrt(x*x+z*z);
    cout<<"Y="<<y;
return 0;
}
```

س/اكتب برنامج لايجاد قيمة  $y$  من المعادلة التالية؟

Y=

**Q3// Write Program (W.P.) to find Y value , when**



```
#include<iostream>
#include<math.h>
using namespace std;
int main()
{
int x; double y;
cin>>x;
if (x%2==0)
    y=sqrt(pow(x,4)+5*x+3);
else
    y=sqrt(pow(x,3)+2*x+5);
cout<<"Y="<<y;
return 0;}
```

**Q0//what is the output of this program:**

```
#include<iostream>
using namespace std;
int main()
{
int x,y,z;
x=y=z=0;
x=++y + ++z;
cout<<x<<y<<z<<endl;
x=y++ + z++;
cout<<x<<y<<z<<endl;
x=++y + z++;
cout<<x<<y<<z<<endl;
x=y-- + --z;
cout<<x<<y<<z<<endl;
return 0;
}
```

Output screen

-----  
**211**

**222**

**533**

**522**

