

BCA 1544



**OBJECT ORIENTED
PROGRAMMING USING C++**

Prepared by,
Aileen Chris.P
AP/BCA

OOPS

- ✓ **Call by Value**
- ✓ **Call by Reference**
- ✓ **Friend functions**

Strings in C++

Inheritance

- ✓ **Template**

Prepared by,
Aileen Chris.P
AP/BCA

CALL-BY-VALUE

- This method copies the value of an argument in to the formal parameter of the subroutine.
- Therefore changes made to the parameter of the subroutine have no effect on the argument used to call it.

Call-by-Reference

- ✓ In this method ,a reference to an argument is passed to the parameter.
- ✓ Inside this subroutine ,this reference is used to access the actual argument specified in the call.
- ✓ This means that changes made to the parameter will affect the argument used to call the subroutine.

Parameters	Call by value	Call by reference
Definition	While calling a function, when you pass values by copying variables, it is known as "Call By Values."	While calling a function, in programming language instead of copying the values of variables, the address of the variables is used it is known as "Call By References."
Arguments	In this method, a copy of the variable is passed.	In this method, a variable itself is passed.
Effect	Changes made in a copy of variable never modify the value of variable outside the function.	Change in the variable also affects the value of the variable outside the function.
Alteration of value	Does not allow you to make any changes in the actual variables.	Allows you to make changes in the values of variables by using function calls.
Passing of variable	Values of variables are passed using a straightforward method.	Pointer variables are required to store the address of variables.
Value modification	Original value not modified.	The original value is modified.
Memory Location	Actual and formal arguments will be created in different memory location	Actual and formal arguments will be created in the same memory location
Safety	Actual arguments remain safe as they cannot be modified accidentally.	Actual arguments are not Safe. They can be accidentally modified, so you need to handle arguments operations carefully.
Default	Default in many programming languages like C++.PHP. Visual Basic NET, and C#.	It is supported by most programming languages like JAVA, but not as default.

FRIEND FUNCTIONS

- A function that has access to the private member of the class but is not itself a member of the class is called friend functions.
- The general form is

```
friend data_type function_name( );
```
- Friend function is preceded by the keyword 'friend'.

PROPERTIES

- Friend function is not in the scope of the class to which it has been declared as friend. Hence it cannot be called using the object of that class.
- Usually it has object as arguments.
- It can be declared either in the public or private part of a class.
- It cannot access member names directly. It has to use an object name and dot membership operator with each member name.

eg: (A . x)

Strings in C++

- **strcpy(s1, s2);**

Copies string s2 into string s1.

- **strcat(s1, s2);**

Concatenates string s2 onto the end of string s1.

- **strlen(s1);**

Returns the length of string s1.

- **strcmp(s1, s2);**

- Returns 0 if s1 and s2 are the same; less than 0 if $s1 < s2$; greater than 0 if $s1 > s2$.

CONTD.,

- **strchr(s1, ch);**

Returns a pointer to the first occurrence of character ch in string s1.

- **strstr(s1, s2);**

Returns a pointer to the first occurrence of string s2 in string s1

EXAMPLE PROGRAM

```
#include <iostream>
#include <cstring>
using namespace std;
int main () {
    char str1[10] = "Hello";
    char str2[10] = "World";
    char str3[10];
    int len ;
    // copy str1 into str3
    strcpy( str3, str1);
    cout << "strcpy( str3, str1) : " << str3 << endl;
    // concatenates str1 and str2
    strcat( str1, str2);
    cout << "strcat( str1, str2): " << str1 << endl;
    // total length of str1 after concatenation
    len = strlen(str1);
    cout << "strlen(str1) : " << len << endl;
    return 0;
}
```

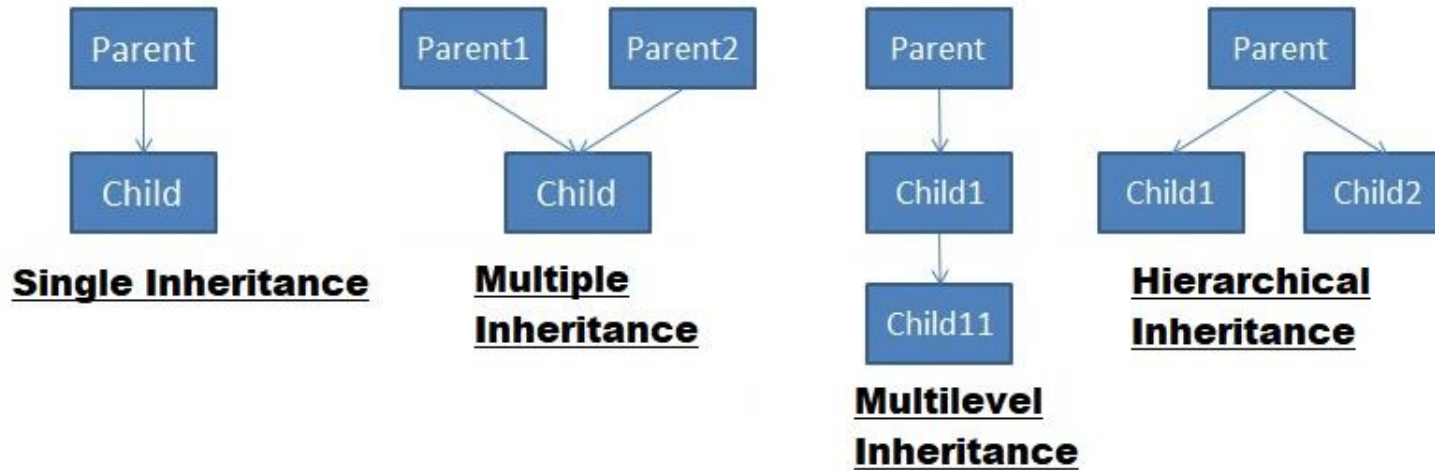
WHAT IS INHERITANCE?

The capability of a class to derive properties and characteristics from another class is called **Inheritance**. Inheritance is one of the most important feature of Object Oriented Programming.

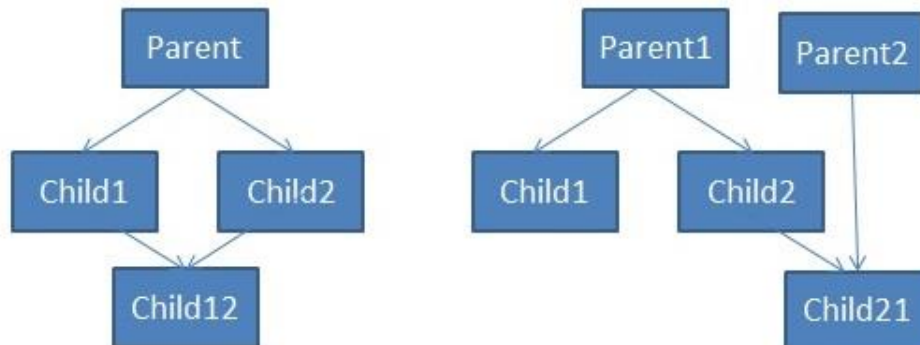
Sub Class: The class that inherits properties from another class is called Sub class or Derived Class.

Super Class: The class whose properties are inherited by sub class is called Base Class or Super class.

INHERITANCE TYPES



Hybrid Inheritance



TEMPLATE

Compiler internally generates and adds below code

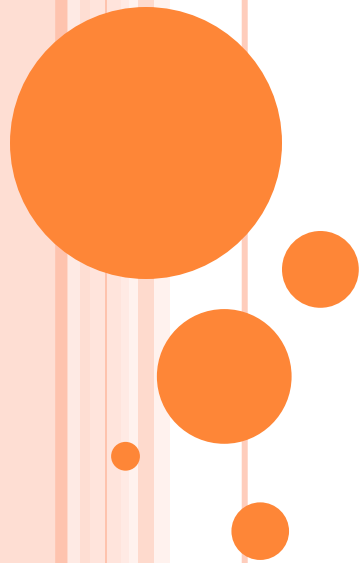
```
template <typename T>
T myMax(T x, T y)
{
    return (x > y)? x: y;
}
```

```
int myMax(int x, int y)
{
    return (x > y)? x: y;
}
```

```
int main()
{
    cout << myMax<int>(3, 7) << endl;
    cout << myMax<char>('g', 'e') << endl;
    return 0;
}
```

Compiler internally generates and adds below code.

```
char myMax(char x, char y)
{
    return (x > y)? x: y;
}
```



THANK YOU !!!