

INHERITANCE

Inheritance is one in which a new class is created that inherits properties of already exist class. It supports concept of code reusability and reduces length of code in object oriented - programming.

Types: 1. Single inheritance 2. Multi-level inheritance
3. Multiple inheritance 4. Hybrid inheritance
5. Hierarchical inheritance.

EXAMPLE PROGRAM (C++)

```
#include <iostream.h>
using namespace <std>;
class A {
    int a, b;
public:
    void add (int x, int y)
    {
        a = x;
        b = y;
        cout << (a+b) << endl; }
class B : public A {
public:
    void print (int x, int y)
    {
        add (x, y); }
int main()
{
    B b1;
    b1.print (5, 6);
}
```

POLYMORPHISM

Polymorphism is that in which we can perform a task in multiple forms or ways. It is applied to the functions or methods. Polymorphism allows object to decide which form of function to implement at compile-time as well as run-time.

EXAMPLE:

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

class A {
    int a, b, c;
public:
    void add (int x, int y)
    {
        a = x;
        b = y;
        cout << "add of a+b is: " << (a+b) << endl;
    }
    void add (int x, int y, int z)
    {
        a = x;
        b = y;
        c = z;
        cout << "add of x+y+z is: " << (a+b+c) << endl;
    }
    virtual void print()
    { cout << "class A's method is running" << endl; }
};

class B: public A {
public: void print()
    { cout << "class B's method is running" << endl; }
};

int main()
{
    A a1;
    a1.add(6, 5);
    B b1;
    b1.print();
}
```