V. Yogeshwaran
192110180

* 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 objects to decide which form of function to implement at complete-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<<"add class A's method is running"<<endl;
};
class B:public A{
public:void main()
{
  cout<<"class B's method is running"<<endl;
}} int main()
{
```

A a1;
a1.add(6,5);
b1.print();

* 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: * "Single inheritance" * "Multi-level inheritance"
* "Multiple inheritance" * "Hybrid inheritance"
* "Hierarchial inheritance"

Example program(PP)

```
#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);{y;
   {add(x,y);};};
int main()
{
  B b1;
  b1.print(5,6);
}
```