```
// C++ Program to demonstrate
// the Virtual Function
#include <iostream>
using namespace std;

// Declaring a Base class
class GFG_Base {

public:
	// virtual function
	virtual void display()
	{
		cout << "Called virtual Base Class function" <<
					"\n\n";
	}

	void print()
	{
		cout << "Called GFG_Base print function" <<
					"\n\n";
	}
};

// Declaring a Child Class
class GFG_Child : public GFG_Base {

public:
	void display()
	{
		cout << "Called GFG_Child Display Function" <<
					"\n\n";
```

```
	}

	void print()
	{
		cout << "Called GFG_Child print Function" <<
					"\n\n";
	}
};

// Driver code
int main()
{
	// Create a reference of class bird
	GFG_Base* base;

	GFG_Child child;

	base = &child;

	// This will call the virtual function
	//base->GFG_Base::display();
base->display();
child.display();
	// this will call the non-virtual function
	base->print();
}


#include <iostream>
using namespace std;
class A {
```

```
public:
  void disp(){
     cout<<"Super Class Function"<<endl;
  }

};
class B: public A{
public:
  void disp(){
     cout<<"Sub Class Function";
  }
};
int main() {
  //Parent class object
  A obj;
  obj.disp();
  //Child class object
  B obj2;
  obj2.disp();
  return 0;
}

#include <iostream>
using namespace std;
class DemoClass {
public:
    int demoFunction(int i) {
        return i;
    }
    double demoFunction(double d) {
        return d;
```

```
    }
};
int main(void) {
    DemoClass obj;
    cout<<obj.demoFunction(100)<<endl;
    cout<<obj.demoFunction(5005.516);
   return 0;
}

#include <iostream>
using namespace std;
class BaseClass {
public:
   void disp(){
      cout<<"Function of Parent Class";
   }
};
class DerivedClass: public BaseClass{
public:
   void disp() {
      cout<<"Function of Child Class";
   }
};
int main() {

  DerivedClass obj;
   obj.disp();
   return 0;
}
```

```
#include <iostream>
using namespace std;
class BaseClass {
public:
  void disp(){
     cout<<"Function of Parent Class";
  }
};
class DerivedClass: public BaseClass{
public:
  void disp() {
     cout<<"Function of Child Class";
  }
};
int main() {

    BaseClass obj;
  obj.disp();
  return 0;
}



// C++ Program to demonstrate
// the Virtual Function
#include <iostream>
using namespace std;

// Declaring a Base class
class GFG_Base {

public:
```

```
	// virtual function
	virtual void display()
	{
		cout << "Called virtual Base Class function" <<
					"\n\n";
	}

	void print()
	{
		cout << "Called GFG_Base print function" <<
					"\n\n";
	}
};

// Declaring a Child Class
class GFG_Child : public GFG_Base {

public:
	void display()
	{
		cout << "Called GFG_Child Display Function" <<
					"\n\n";
	}

	void print()
	{
		cout << "Called GFG_Child print Function" <<
					"\n\n";
	}
};
```

```
// Driver code
int main()
{
	// Create a reference of class bird
	GFG_Base* base;

	GFG_Child child;

	base = &child;

	// This will call the virtual function
	//base->GFG_Base::display();
base->display();
child.display();
	// this will call the non-virtual function
	base->print();
}
```