```
#ifndef BASE_H
#define BASE_H

class Base
{
public:
        Base (int = 0);

        void Display () const;
        virtual void Print () const;

private:
        int x;
};

#endif
```

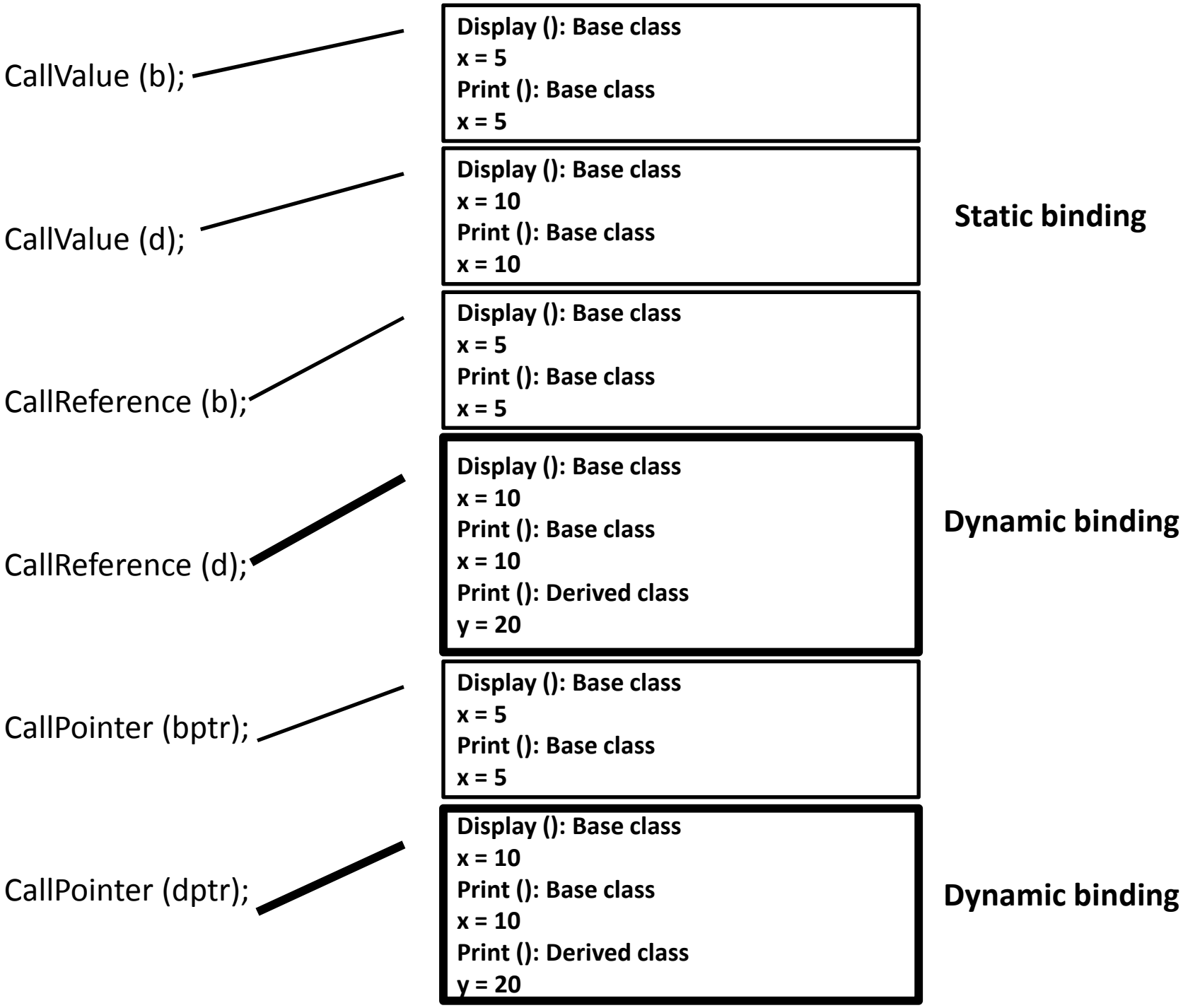
```
#include <iostream>
using namespace std;

#include "Base.h"

Base::Base (int a)
: x (a)
{
}

void Base::Display () const
{
        cout<<"Dispaly (): Base class\n x =
"<<x<<endl;
}

void Base::Print () const
{
        cout<<"Print (): Base class\n x =
"<<x<<endl;
}
```

```
#ifndef DERIVED_H
#define DERIVED_H

#include "Base.h"

class Derived : public Base
{
public:
        Derived (int = 0, int = 0);

        void Display () const;
        virtual void Print () const;

private:
        int y;
};

#endif
```

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

#include "Derived.h"

Derived::Derived (int a, int b)
: Base (a), y (b)
{
}

void Derived::Display () const
{
        Base::Display ();
        cout<<"Dispaly (): Derived class\n y =
"<<y<<endl;
}

void Derived::Print () const
{
        Base::Print ();
        cout<<"Print (): Derived class\n y =
"<<y<<endl;
}
```

```
#include "Base.h"
#include "Derived.h"

void CallValue (Base b);
void CallReference (Base& b);
void CallPointer (Base* b);

void main ()
{
        Base b (5);
        Derived d (10, 20);

        CallValue (b);
        CallValue (d);

        CallReference (b);
        CallReference (d);

        Base* bptr = new Base (5);
        Derived* dptr = new Derived  (10, 20);

        CallPointer (bptr);
        CallPointer (dptr);
}
```

```
void CallValue (Base b)
{
        b.Display ();
        b.Print ();
}

void CallReference (Base& b)
{
        b.Display ();
        b.Print ();
}

void CallPointer (Base* b)
{
        b->Display ();
        b->Print ();
}
```


CallValue (b);
Display (): Base class
x = 5
Print (): Base class
x = 5
CallValue (d);
Display (): Base class
x = 10
Print (): Base class
x = 10
Static binding
CallReference (b);
Display (): Base class
x = 5
Print (): Base class
x = 5
CallReference (d);
Display (): Base class
x = 10
Print (): Base class
x = 10
Print (): Derived class
y = 20
Dynamic binding
CallPointer (bptr);
Display (): Base class
x = 5
Print (): Base class
x = 5
CallPointer (dptr);
Display (): Base class
x = 10
Print (): Base class
x = 10
Print (): Derived class
y = 20
Dynamic binding