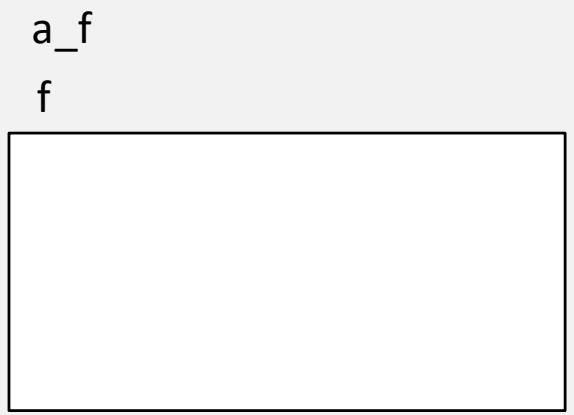
THE const KEYWORD

Used With Classes

Delroy A. Brinkerhoff

PASS-BY-REFERENCE

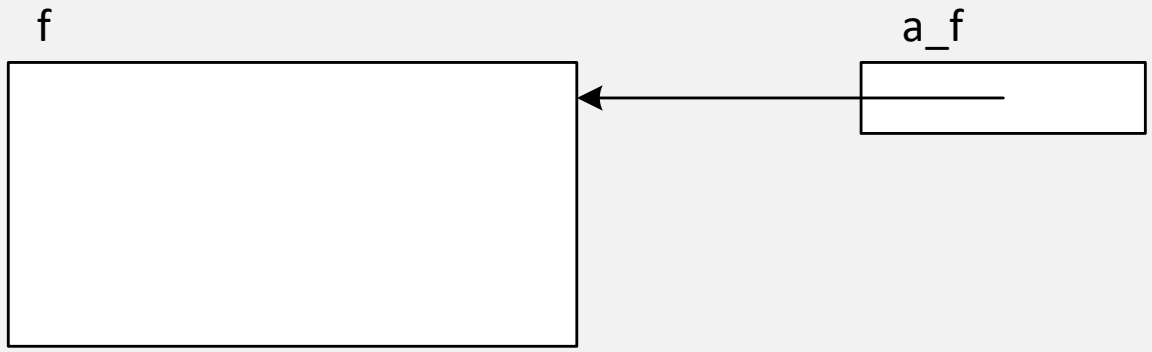
- `b.function1(f);`
- `void function1(const Foo& a_f);`



PASS-BY-POINTER

- `b.function2(&f);`
- `b.function3(&f);`
- `b.function4(&f);`

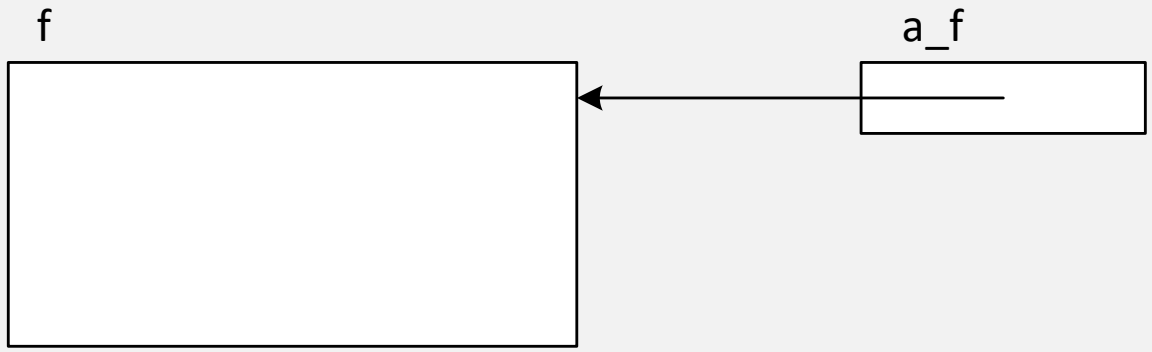
- `void function2(const Foo* a_f);`
- `void function3(Foo* const a_f);`
- `void function4(const Foo* const a_f);`



PASS-BY-POINTER

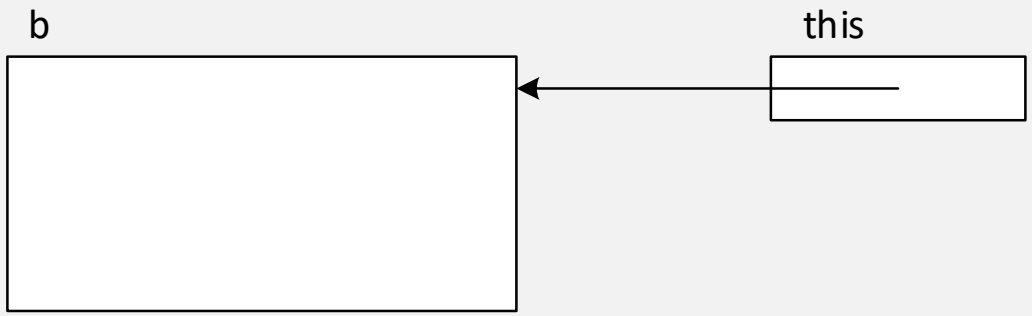
- `b.function2(&f);`
- `b.function3(&f);`
- `b.function4(&f);`

- `void function2(Foo const* a_f);`
- `void function3(Foo* const a_f);`
- `void function4(Foo const* const a_f);`



IMPLICIT / "THIS" OBJECT: PASS BY POINTER

- `b.function2();`
- `void function2() const;`



GETTERS: JAVA

- int height;
- double weight;
- String name;

- Other class references
  - implements cloneable
  - clone

```
int getHeight()
{
        return height;
}
```

```
String getName()
{
        return name;
}
```

GETTERS: C++

- int height;
- double weight;
- string name;

- Other class pointers and references
  - const return

```
int getHeight()
{
        return height;
}
```

```
string getName()
{
        return name;
}
```

GETTERS: CONST RETURN

CLASS

- `char name[100];`

- ```
  const char* get_name()
  {
    return name;
  }
  ```

CLIENT

- `const char* student = p.get_name();`
- `char const* student = p.get_name();`

- ~~`char* const student = p.get_name();`~~
  ```

SYMBOLIC CLASS CONSTANTS

DEFINING

```
class foo
{
  public:
    const static int N = 10;
    void function();
};
```

USING

- ```
  void foo::function()
  {
     ... N ...
  }
  ```
- ```
  void application1()
  {
     ... foo::N ...
  }
  ```