

Delegates

Doug Blank
Programming Paradigms
C# - Spring 2010

Functions as Firstclass Objects in Python

```
def plus(i, j):  
    return i + j  
def mult(i, j):  
    return i * j
```

```
x = plus  
x(6, 7)
```

```
for func in [mult, plus]:  
    print func(20, 6)
```

Functions as Firstclass Objects

- Functions can be:
 - Arguments
 - Values of Variables
 - Return Values
 - Stored in data structures
- Higher Order Functions (HOF)
 - Functions that take functions as arguments
- Full-fledged objects

Functions in C

```
#include <stdio.h>
```

```
int plus(int i, int j) {  
    return i + j;  
}
```

```
int mult(int i, int j) {  
    return i * j;  
}
```

```
int main(int argc, char *argv []) {  
    printf( "Plus: %d\n", plus(3, 6) );  
}
```

Functions Pointers in C

```
#include <stdio.h>

int plus(int i, int j) {
    return i + j;
}
int mult(int i, int j) {
    return i * j;
}
int apply(int (*func)(int, int), int i, int j) {
    return (*func)(i, j);
}
int main(int argc, char *argv []) {

    printf( "Apply Plus: %d\n", apply(plus, 3, 6) );

}
```

C# Example

```
using System;

class Example {

    public static int Plus(int i, int j) {
        return i + j;
    }

    public static int Mult(int a, int b) {
        return a * b;
    }

    public static void Main() {
        Console.WriteLine("Plus: {0}", Plus(4, 5));
        Console.WriteLine("Mult: {0}", Mult(4, 5));
    }
}
```

C# Example of “Function Pointer”

```
using System;
```

```
delegate int Function(int i1, int i2);
```

```
class Example {  
    public static int Apply(Function f, int i1, int i2) {  
        return f(i1, i2);  
    }  
    public static int Plus(int i, int j) {  
        return i + j;  
    }  
    public static int Mult(int a, int b) {  
        return a * b;  
    }  
    public static void Main() {  
        Console.WriteLine("Apply Plus: {0}", Apply(Plus, 4, 5));  
        Console.WriteLine("Apply Mult: {0}", Apply(Mult, 4, 5));  
    }  
}
```

Functions and Names in Scheme

```
(define plus  
  (lambda (i j)  
    (+ i j)))
```

```
(plus 5 8)
```


Lambda in Python

```
func = lambda x, y: x + y
```

```
func(5, 6)
```

Lambda in C#

```
using System;

class Example {

    public static void Main() {

        // C# 2.0:
        listOfFoo.Where(delegate(Foo x) { return x.Size > 10; });

        // C# 3.0:
        listOfFoo.Where(x => x.Size > 10);

    }
}
```