



CSCI 2270 – CS 2: Data Structures



University of Colorado
Boulder

Reminders



Topics

- Parameter Passing
 - Pass by Value
 - Pass by Reference
 - Pass by Array
 - Pass by Pointer



Parameter Passing

- Pass by Name
 - Algol60
- Pass by Value-Result
 - Fortran, sometimes Ada
- Pass by Value
 - C/C++, Java, Pascal, Ada
- Pass by Reference
 - Fortran, Pascal
- Pass by Array
 - Technically not one of the official parameter passing methods
- Pass by Pointer
 - Technically not one of the official parameter passing methods



Pass by Value

```
void increment(int a, int b)
{
    a += b;
    cout << a << " " << b << endl;
}
```

```
int main(void)
{
    int x = 3, y = 7;

    increment(x, y);
    cout << x << " " << y << endl;
    return 0;
}
```

Note: x and y are unaffected in main()



Pass by Reference

```
void func(int& x, int& y)
{
    x = 3;
    y = 4;
}
```

```
int main()
{
    int a = 1, b = 2;
    cout << "Before func()\n";
    cout << a << " " << b << endl;    // 1 2

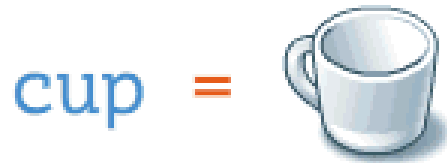
    func(a, b);

    cout << "After pass by reference\n";
    cout << a << " " << b << endl;    // 3 4
}
```



Pass by Reference vs Value

pass by reference



fillCup()

pass by value



fillCup()

www.mathwarehouse.com



Pass by Array

- See Week 1 lecture for examples



Pass by Pointer

```
void func(int *x, int *y)
{
    *x = 3;
    *y = 4;
}
```

```
int main()
{
    int a = 1, b = 2;
    cout << "Before func()\n";
    cout << a << " " << b << endl;    // 1 2

    func(&a, &b);

    cout << "After pass by reference\n";
    cout << a << " " << b << endl;    // 3 4

    return 0;
}
```



Pass by Pointer

```
void func(int *x)
{
    *x = 3;
}
```

```
int main()
{
    int a = 1;
    int *ptr = &a;

    cout << a << endl;    // 1

    func(ptr);

    cout << a << endl;    // 3

    return 0;
}
```



Pass Pointer by Reference

```
int global = 100;

// change ref to ptr
void func(int *& x)
{
    x = &global;
}
```

```
int main()
{
    int var = 3;
    int *ptr_to_var = &var;

    cout << "Before :" << *ptr_to_var << endl; // 3

    func(ptr_to_var);

    cout << "After :" << *ptr_to_var << endl; // 100

    return 0;
}
```



Questions?

