The Art of Identifier Naming

Jonathan I. Maletic
Kent State University

Names

- Variables
- Types, classes
- Methods, functions
- Global constants

The most difficult part of programming is coming up with good variable names.

Describe this code:

```
int f(int x, int y) {
  int z;
  z = x * y;
  return z;
}
```

Describe this code:

```
int f(int h, int w) {
  int a;
  a = h * w;
  return a;
}
```

Describe this code:

```
int areaOfRectangle(int height, int width) {
  int area;
  area = height * width;
  return area;
}
```

Nouns

- Variables, objects, types, classes
  - Person, stack, list, window, menu
- Plurals should be avoided
  - String people[n]; // people[3]="joe";
  - String person[n]; // person[3]="joe";

Verbs

- functions, methods
  - GetValue, push, computeSalary
  - DrawWindow, update
  - SizeOf, add, isEqual

Naming style

- Consistent style should be used
- camelCase - capitalize words (after the first)
- under_score - underscores between words.

Indentation & Spacing

- Use consistent indentation and spacing
- Most IDEs use 4 spaces per block
- Take advantage and use the auto indent
- Take time to go back and clean up indentation

Ugly & Unreadable

```
 void foo(int x) {
int   t;
 for ( i=0;    i<  x; ++i) {


cout << i;
t =t+  i; }
 cout     <<  t;
  }
```

Clean & Neat

```
void foo(int x) {
    int   t;
    for (i=0; i< x; ++i) {
        cout << i;
        t = t +  i;
    }
     cout <<  t;
}
```