

## Variables in Calling Function vs Variables in Called Function

- absolutely have NOTHING to do with each other
  - even if their names are identical
- main function, ralph function compiled *completely separately* (possibly on different days)
  - at execution time, main calls ralph
  - at their individual compile times, **completely separate** codes
  - when both are in one file which is "compiled", **TWO** separate compilations take place

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

```
void ralph(int x);
```

```
main( )
```

```
{
```

```
    int var_one, var_two;          // LOCAL to main !!!!
```

```
    var_one = 3;
```

```
    var_two = 4;
```

```
    printf("First output (in main, var_one, var_two): %d %d \n\n",  
           var_one, var_two);
```

```
    ralph(var_two);
```

```
    printf("Fourth output (in main, var_one, var_two): %d %d \n\n",  
           var_one, var_two);
```

```
}
```

```
void ralph(int var_three)          // value in var_two passed to var_three
```

```
{
```

```
    int var_one, var_two;          // LOCAL to ralph !!!!
```

```
    printf("Second output (in ralph, the passed value in var_one): %d \n\n",  
           var_three);
```

```
    var_one = var_three * 2; // change LOCAL var_one
```

```
    var_two = 95;
```

```
    printf("Third output (in ralph, var_one var_two both belonging  
           " to ralph): %d %d \n\n", var_one, var_two); // separate issue
```

```
}
```

First output (in main, var\_one, var\_two): 3 4

Second output (in ralph, the passed value in var\_one): 4

Third output (in ralph, var\_one var\_two both belonging to ralph): 8 95

Fourth output (in main, var\_one, var\_two): 3 4

Press Enter to return to Quincy...