

**Declare a local variable
where it logically belongs**

Generally, close to its first use

```

/** Sort array segment b[0..n] */
public void selectionSort(int[] b, int n) {
    int temp;
    int j;
    // inv: b[0..k-1] is sorted & b[0..k-1] <= b[k..n]
    for (int k=0; k < n; k= k+1) {
        // Set j to the index of the minimum of b[k..n]
        j= k;
        for (int p= k+1; p<= n; p= p+1) {
            if (b[p] < b[j]) j= p;
        }
        // Swap b[j] and b[k]
        temp= b[j]; b[j]= b[k]; b[k]= temp;
    }
}

```

inv: b

0	sorted <=	k	>=	n
---	-----------	---	----	---

CS101J, Cornell 0

**Declare local variables
where they logically belong**

```

/** Sort array segment b[0..n] */
public void selectionSort(int[] b, int n) {
    int j;
    // inv: b[0..k-1] is sorted & b[0..k-1] <= b[k..n]
    for (int k=0; k < n; k= k+1) {
        // Set j to the index of the minimum of b[k..n]
        j= k;
        for (int p= k+1; p<= n; p= p+1) {
            if (b[p] < b[j]) j= p;
        }
        // Swap b[j] and b[k]
        int temp= b[j]; b[j]= b[k]; b[k]= temp;
    }
}

```

inv: b

0	sorted <=	k	>=	n
---	-----------	---	----	---

CS101J, Cornell 1

**Declare local variables
where they logically belong**

```

/** Sort array segment b[0..n] */
public void selectionSort(int[] b, int n) {
    // inv: b[0..k-1] is sorted & b[0..k-1] <= b[k..n]
    for (int k=0; k < n; k= k+1) {
        // Set j to the index of the minimum of b[k..n]
        int j= k;
        for (int p= k+1; p<= n; p= p+1) {
            if (b[p] < b[j]) j= p;
        }
        // Swap b[j] and b[k]
        int temp= b[j]; b[j]= b[k]; b[k]= temp;
    }
}

```

inv: b

0	sorted <=	k	>=	n
---	-----------	---	----	---

CS101J, Cornell 2