Class 15
Reference Parameters
Example 1

```c
int product(int a, int b){
    a = a * b;
    return a;
}

int main(){
    int x = 5, y = 6;
    cout << product(x, y) << endl;
    cout << x;
    return 0;
}
```
Call by value

• When passing values to a function, C++ creates a copy of the values stored in the variable

• The function operates on those copies of values
Example 2

```c
void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}

int main() {
    int x = 5, y = 3;
    swap(x, y);
    cout << "x = " << x << " ; y = " << y << endl;
    return 0;
}
```
Call by reference

• When you want to pass the actual variable to the function, you mark this in the title line by putting an & between the type and name of the parameter
Example 3

```c++
void swap(int &a, int &b){
    int temp = a;
    a = b;
    b = temp;
}

int main(){
    int x = 5, y = 3;
    swap(x, y);
    cout << "x = " << x << " y = " << y << endl;
    return 0;
}
```
Example 4

```cpp
void applyCurve(int &score){
    score = score + 10;
}

int main(){
    int grade = 75;
    applyCurve(grade);
    cout << grade << endl;
    return 0;
}
```
Key summary

• Call by value parameter:
  • A copy of the value is passed
  • Changes made to the value inside the function are not permanent
  • An argument can be a hard-coded number, for example:
    • $\sqrt{5.0}$;

• Call by reference parameter:
  • Changes are permanent
  • A call by reference argument must be a variable