return 0;
}

// function template definition
template <class Any> // or typename Any
void Swap(Any &a, Any &b)
{
    Any temp; // temp a variable of type Any
    temp = a;
    a = b;
    b = temp;
}

The first Swap() function has two int arguments, so the compiler generates an int version of the function. That is, it replaces each use of Any with int, producing a definition that looks like this:

void Swap(int &a, int &b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}

You don’t see this code, but the compiler generates, then uses it in the program. The second Swap() function has two double arguments, so the compiler generates a double version. That is, it replaces Any with double, generating this code:

void Swap(double &a, double &b)
{
    double temp;
    temp = a;
    a = b;
    b = temp;
}