COMPUTATIONAL MATH, SCIENCE AND ENGINEERING DEPARTMENT	MICHIGAN STATE UNIVERSITY
	389C7E8
0020 PROCUENT COSHAGO ESGADOS/N VIDERRO PROCUENT COSSHAGO ESHAGORU VERRERO VERRERO MARKETON COSSHAGO ERFETTE PER COMPARTO EN 6592 (488554 880000000 000000000 808A0100 00004889 45F08955 F8DBGDF0 5DC35F54 8955467 COFFFFFF FFBAFETT 000004889 45F08955 F8DBGDF0 5DC3FFC5 20 6655 (58257466 0000755 2825620 5725766 00005575 27256200 5573786 00005575 28256200 577578 2825620 577578 28256	250C355 . n.z.jyonaka.j n.z.jyonaka.j
4839548 88000000 00000000 80BA0100 00004889 45F08955 F8DB0DF0 5DC35548 89E548C7 COFFFFFF FBAFE7F 00004889 45F08955 F8DB0DF0 5DC35F25 21 6656 FF252A06 00000000000000000000000000000000000	70 FF55 . "%9 . "%B . "%F . "%H . "%J . "%L . "%N . "%P . "%R . "%
0784 54060000 4C8DID95 05000041 53FF2585 05000090 \$0000000 00E96FF FFF6819 000002E9 \$FFFFF 68250000 00E90ZFF FFF6830 000000E9 C8FFFFF 68470000 00E96FF FFF6887 000000E9 8CFFFFF 6870000 00E96FF FFF6887 000000E9 8CFFFFF 6870000 00E96FF FFF6887 000000E9 8CFFFFF 6870000 00E96FF FFF6887 000000E9 8CFFFFFF 6870000 00E96FF FFF6887 00000E9 8CFFFFFF 6870000 00E96FF FFFF6887 00000E9 8CFFFFFF 6870000 00E96FF FFFFFF 6870000 00E96FF FFFFFFFF 6870000 00E96FF FFFFFF 6870000 00E96FF FFFFFFF 6870000 00E96FF FFFFFFFF 6870000 00E96FF FFFFFFF 6870000 00E96FF FFFFFFFF 6870000 00E96FF FFFFFFF 687000 00E96FF FFFFFFF 687000 00E96FF FFFFFFF 687000 00E96FF FFFFFFFFF 687000 00E96FF FFFFFFFF 687000 00E96FF FFFFFFFFF 687000 00E96FF FFFFFFFFFF 687000 00E96FF FFFFFFFFFFF 687000 00E96FF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	34F0000 1Lç.iAS %0enEE hE< h+E" h=E» h0 DE982FF .Èœ¯¯haÈ¥⊤¯hsÈ™¯ThÖȆ¯ThóÈn⊤¯h∑Èå¯Th¯Èǯ
6912 FFFF0000 00000000 FFFF7F7F FFFF7FFF 53697A65 206F6620 73686F72 743A0053 6D616C6C 65737420 73686F72 743A004C 61726765 73742073 686F7274 3/	1005369 **
6976 7A65206F 6620696E 743A0053 6D616C6C 65737420 696E743A 004C6172 67657374 20696E74 3A005369 7A65206F 66206C6F 6E673A00 536D616C 6C657374 20	
7104 666C6F61 743A004C 61726765 73742066 6C6F6174 3A004469 67697473 20696E20 6D617469 7373612C 20666C6F 61743A00 53697A65 206F6620 646F7562 60	
7168 5360616C 6C657374 20646F75 626C653A 004C6172 67657374 20646F75 626C653A 00446967 69747320 696E206D 61746973 73612C20 646F7562 6C653A00 53	
7232 206F6620 6C6F6E67 20646F75 626C653A 00536D61 6C6C6573 74206C6F 6E672064 6F75626C 653A004C 61726765 7374206C 6F6E6720 646F7562 6C653A00 00	
7296 44696769 74732069 6E206D61 74697373 612C206C 6F6E6720 646F7562 6C653A00 14000000 00000000 017A5200 01781001 100C0708 90010000 340000000 10	
	00000004le********
	000000D ÜÜ
	00000004,D*******
	10000000
	10000000 \$UUJ/eU44
	3000000 /%Ó
CMSE 822, FS21, W.F. Punch	
3576 82000000 00000000 00000000 00000000 000000	300000

MICHIGAN STATE

overloaded function

We've seen this before. An overloaded function is a function that

- has one name
- represents different operations depending on its parameter types

C++ supports function overloading

MICHIGAN STATE UNIVERSITY

name mangling

Real process, how the compiler creates a unique name based on the function name and its associated types.

- mangled name allows for look up of the correct function
 - nm shows mangled names
 - http://demangler.com/



function signature

Function signature consists of:

- function name
- function return type
- the types, and their order, of the parameters

Names of the parameters do not matter!

Uniquely identifies (or should) a function

Сомрі	UTATIONAL MATH, SCIENCE AND E	NGINEERING DEPARTMENT	MICHIGAN STATE UNIVERSITY
	00 488B15C5 0A000048 89D64889 C7E8EC04 0000E8CF 04000048	89C3488D 351E0600 00488B05 9C0A0000 4889C7E8 04050000 4889DE48 89C7E8E1 040000 89C7E89F 04000048 8035E705 0000488B 05570A00 004889C7E 88F6400 00BE0300 000044	248Hã.≈Há÷Há«ËÏËœHà√Hç5Hã.úHá«ËHàfiHá«Ë∙.
	04 0000488B 15430A00 004889D6 4889C7E8 6A040000 488B1533		000 «Ë†Hā.CHà÷Hà«ËjHã.1Hà÷Hà«ËXHç5∑HãHà«Ëx
			348 n√Hà«ËHä.øHà÷Hà«ËÊ˱f.~√Hç5[Hä.ïHå«Ë"f.
	48 8B150609 00004889 D64889C7 E82D0300 00E8E602 00006648 8B 15C70390 004889D6 4889C7E8 EE0200 30 E841070 7066480F	. 0F7EC348 8D35DA04 0000488B 05DB0800 004889C7 E8430300 0066480F 6EC34889 C7E80 7EC3488D 35AC0400 00488B07 C080000 4889C7E8 04030000 66480F6E C34889C7 E8C70	503HäHå÷Hå«ËËËfH.~√Hç5/Hä.€Hå«ËCfH.n√Hå« 200Hä.«Hå÷Hå«Ëó˰,⊋fH.~√Hc5''Hä.úHå«ËfH.n√Hå«Ë
		rent functions v	
	00 4889D048 89C/E833 020000E8 F8010000 Ubil D04 8D353504	00004668 05E50700 00488 C7 E6450200 00050000 DB5C446 05C7E513 02000048 8BLSC	167Ha-Ha«E3E1€)-Hç53Ha.,Hå«EK⊕n-∈≤Hå«EHä
			200nu÷nu«ctc+ey/nç3nu.♥nu«cen/e <pnu«cnu.e< td=""></pnu«cnu.e<>
	same nan	1 000004889 C/E8AC01 000004889 C/E8AC01 000004888 15580/00 000488916 4889C/E8 820100 1 0 0 0000885 FFFFFSD C3554889 ES88080 FFFFSDC3 554889ES B8FF7F00 005DC	555 HÉfH[]/UHÁÁHÉÏ.a}, au É}, .u2Å}****u)Hç=**Ë}HçÁ**H 255 Hā
	00 4C8D1D95 05000041 53FF2585 05000090 68000000 00E9E6FF		000 TLç.ïAS [*] %ÖêhÈÊ ^{****} h+È ^{*****} h=È» ^{****} h
		. 000000E9 A0FFFFFF 68970000 00E996FF FFFF68B7 000000E9 8CFFFFFF 68F70000 00E98: 743A0053 6D616C6C 65737420 73686F72 743A004C 61726765 73742073 686F7274 3A005	
		67657374 20696E74 3A005369 7A65206F 66206C6F 6E673A00 536D616C 6C657374 206C6	F6E ze of int:.Smallest int:.Largest int:.Size of long:.Smallest
	.6C 6C657374 20646F75 626C653A 004C6172 67657374 20646F75	626C653A 00446967 69747320 696E206D 61746973 73612C20 646F7562 6C653A00 53697	A65 Smallest double:.Largest double:.Digits in matissa, double:.S
		66672064 6F75626C 653A004C 61726765 7374206C 6F6E6720 646F7562 6C653A00 00000 6C653A00 14000000 00000000 017A5200 01781001 100C0708 90010000 34000000 1C0001	
			FFF i,
			E10Ü
			20D ÜÜÜÜÜÜÜ
			200Ü
			000 4§="""
			000Ü
			E10
			CO7É`4Ù4********HÜC
			3004,D*******ÜÜ
			200 øuöJ?eU4
			000 /
		CMSE 822, FS21, W.F. Punch	
		00000000 00000000 00000000 00000000 0000	

MICHIGAN STATE

two different functions (Ex 6.5)

```
void swap (double &d1, double &d2) {
  cout << "This must be the double swap"<<endl;</pre>
  double temp;
  temp = d1;
  d1 = d2;
  d2 = temp;
                     void swap (int &i1, int &i2) {
                        cout << "This must be the int swap"<<endl;</pre>
                        int temp;
                        temp = i1;
                        i1 = i2;
                        i2 = temp;
```

MICHIGAN STATE

resolving can be complicated

You can find various refs (https://en.cppreference.com/w/cpp/language/overload_resolution) as to the rules.

 the problem is basically conversion.
 What happens if a conversion is available that might convert one type to another?

```
int f() {
                                               Ex 6.6
  cout << "f, no arg"<<endl;</pre>
int f(int i) {
  cout << "f, 1 int arg"<<endl;</pre>
int f(int i,int j){
  cout << "f, 2 int arg"<<endl;</pre>
int f(double x, double y=3.14159) {
  cout << "F, 2 arg with default}"<<endl;</pre>
int main () {
  f(5.65); // which one???
  f(42, 2.65); // which one???
```

MICHIGAN STATE

Easier to have happen then you think

This seems like a bad place to end up, but because code can be written in pieces by different people, conversion functions might creep in that allow for this kind of problem.

Beware!

MICHIGAN STATE

A word on const Trying to differentiate parameter types based on top-level const does not work. These are *the same functions*!

```
long my_fun (const long p1){
  cout << "const fn"<<endl;
}
  int main(){
     const long c_long = 1;
     long my_fun(long p1){
     cout << "reg fn" <<endl;
}

int main(){
     const long c_long = 1;
     long my_long = 2;
     my_fun(c_long);
     my_fun (my_long);
}</pre>
```

COMPUTATIONAL MATH, SCIENCE AND ENGINEERING DEPARTMENT	MICHIGAN STATE UNIVERSITY
0000488B 15470800 004889D6 4889CF8 6E050000 488B1535 0B000048 89D64889 CFESCC5 0000488D 35790600 00488B05 140B0000 488 5376 004889C7 E85D5500 00488B15 000B0000 4889D648 89CFE827 050000E8 10050000 4889C348 B354A06 0000488B 05D70A00 00488BC5 140B0000 488 5376 1059000 488B15C5 0A000048 9D64889 CFESC04 0000E8CF 0400048 95C3488D 351E0600 00488B05 9CA00000 4889CFE8 04050000 488 5504 3B158A0A 0000488B 15430A00 00488B15 780A0000 488B15 780A0000 488B1531 0A000048 8D35E705 0000488B 05570A00 00488B05 100 5682 6C7E8A004 0000488B 15430A00 00488B15 FC090000 4889CFE8 6A040000 488B1531 0A000048 8D64889 CFE85804 0000488B 05570A00 00488B05 100 5682 BE040000 004839CF E8590400 00488B15 FC090000 4889CFE8 6A040000 488B1531 0A000048 8D64889 CFE85804 0000488B 05570A00 00488B05 100 5682 BE040000 004839CF E8590400 00488B15 FC090000 4889CFE8 6A040000 488B1531 0A000048 8D64889 CFE85804 0000488B 0550000 488B055 09000048 8D55280 0000488B 05570A00 00488B05 100 5682 BE040000 0048389 CFE8A040 0000488B 158F0900 00488B0 4889CFE8 E6030000 0EB0100 00666FFC 3488D35 80000048 8D55280 00000488 8D55280 00000488 05570A00 00488B05 100 58760 89C7E8CD 030000048 8B158209 0000488B D64889CF E8A90300 00488B05 1600000 488B05 09000048 8D57280 00000488 057880 00000000 488B05 090000048 8D57280 00000000 0488B05 00000000 488B05 00000000 488B05 00000000 00000000 0000000000 00000000	8F0500 004889DE 4889C7E8
6336 B8000000 004483C4 485BSDC3 55488 55 4883EC10 897DFC39 75F8837D FC017532 817DF8FF FF000075 29488D3D F8070000 E87D0100 004 6400 07000048 8805F60 00004889 C7586C01 0000C9C3 554889E BEFFFF0 00BF0100 0000E8A FFFFFFD C3554889 E588088 FFFFFF5D C3554889 E588088 FFFFFFFF FFFF75D C3554889 E588088 00000000 00000885 DC3554889 E588088 FFFFFFFFF FFFF7F5D C3554889 E588088 000000000 00000885 DC3554889 E588088 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	188D15 04E7FFFF 488D35E5 ∏HE∱H[]√UHÁAHEI.Å], du E}, .u2A}u)H(=E},H(-, A. 'HÇSA 1889ES B8FF7F00 005DC355HaHa«Ēl,VUHÁÀœ'øE®']_VUHÁÄ∏]√UHÁÄ∏]√UHÁÄ∏J\V 554889 ESB8589 01000066 HÁÄ∏Ä]√UHÁÄ∏]√UHÁÄ∏]√UHÁÁM∏
5922 4889E548 88000000 000000000 8080100 00004889 45768955 F8BBGDF0 5DC35548 89E548C7 C0FFFFFF FFRAFE7F 00004889 45F08955 F8BBGDF0 5DC35548 89E548C7 C0FFFFFF FFRAFE7F 00004889 45F08955 F8BGDF0 5DC35548 89E548C7 C0FFFFFF F8FAFE7F 00004889 45F08955 F8BGDF0 5DC35548 89E548C7 C0FFFFFFF 00004889 45F08955 F8BGDF0 5DC35548 89E548C7 C0FFFFFF 00004889 45F08955 F8BGDF0 5DC35548 89E548C7 C0FFFFFFF 00004889 45F08955 F8BGDF0 5DC35548 89E548C7 C0FFFFFF 00004889 45F08955 F8BGDF0 5DC35548 89E548C7 C0FFFFFF 00004889 45F08955 F8BGDF0 5DC35548 89E548C7 C0FFFFFF 00004889 45F08955 89E548 89E	tion for multiple
6848 00E9BEFF FFFF6861 000000E9 B4FFFFFF 68730000 00E9AAFF FFFF6885 000000E9 A0FFFFF 68970000 00E996FF FFFF68B7 000000E9 8CF	FFFFFF 68F70000 00E982FF .Eœ™haE¥™hsE™™hÓE†™hóEñ™hóEå™h⊤EC™
7040 673A004C 61726765 7374206C 6F6E673A 0053697A 65206F66 206C6F6E 67206C6F 6E672069 6E743A00 53697A65 206F6620 666C6F61 743	3AU053 6D616C6C 65737420 g:.Largest long:.Size of long long int:.Size of float:.Smallest
7168 5360616C 6C657374 20646F75 626C653A 004C6172 67657374 20646F75 626C653A 00446967 69747320 696E206D 61746973 73612C20 646	5F6620 646F7562 6C653A00 float:.Largest float:.Digits in matissa, float:.Size of double:. 5F7562 6C653A00 53697A65 Smallest double:.Largest double:.Digits in matissa, double:.Size
7232 206F6620 6C6F6E67 20646F75 626C653A 00536D61 6C6C6573 74206C6F 6E672064 6F75626C 653A004C 61726765 7374206C 6F6E6720 646 7296 44696769 74732069 6E206D61 74697373 612C206C 6F6E6720 646F7562 6C653A00 14000000 00000000 017A5200 01781001 100C0708 900	
7296 44696769 74732069 6E206D61 74697373 612C206C 6F6E6720 646F7562 6C653A00 14000000 00000000 017A5200 01781001 100C0708 900 7360 69FCFFFF FFFFFFF 0B000000 00000000 00000000 00000000	7000000 35F5FFF FFFFFF L,
7552 8620403 0000000 0004000 0000000 00000000 54000000 FC000000 BSFBFFFF 10000000 00000000 00040100 000	200E10 86020403 00000000 Ü
	000C07 08000000 00000000041e*******ÜÜ
	000000 34000000 DC010000 4\$=""
	000000 00040100 00000E10
	00000D 06044300 00000C07É.`4Ù4 ^{******} HÜÜC
8320 29190000 01000000 941A0000 01000000 9E1A0000 01000000 A81A0000 01000000 B21A0000 01000000 BC1A0000 01000000 C61A0000 010	
8384 DA1A0000 01000000 E41A0000 01000000 EE1A0000 01000000 F81A0000 01000000 14190000 01000000 00000000 00000000 00000000	200000 00000000 00000000 /%Ó"
CMSE 822, FS21, W.F. Punch	
8576 00000000 00000000 00000000 00000000 0000	180000 00000000 08000000

MICHIGAN STATE

overloading, double edge sword

Nice to be able to overload a function based on types

What a pain, some function (very general) requires that I re-write it for every type, especially for any new one I create!

MICHIGAN STATE

template

The way to get around it is called a template. A template is a *pattern*, a pattern that can be used to *create a function* with whatever types we want.

Need to get that a <u>template is not a</u> <u>function</u>, it is how to create a function with some type information set

MICHIGAN STATE UNIVERSITY

basis of everything in the STL

While pointers are a basis for a lot of how C (the underlying language) works, templates are the basis for C++/STL and how it really solves many problems of generality with types.

Ex 6.7

```
template <typename my_type>
void swap (my_type &first, my_type &second) {
   my_type temp;
   temp = first;
   first = second;
   second = first;
}
```

```
type in-between
              < >
                           template type
        keyword
                           variable
            ypename my_type
void swap (my_type &first, my_type &second) {
  my_type_temp;
  temp = first;
  first = second;
                          use the template type
                          var everywhere
  second = temp;
                          you need template
                          behavior
```

```
template <typename my type>
void swap (my type& first,
my type& second) {
                                   1) look for swap
                                   with two ints
  my type temp;
  temp = first;
                                       int i=1, j=2;
   first = second;
                                       swap(i,j);
   second = temp;
                                 3. Call
                                 new fn
  2) substitute
                        void swap (int& first, int&
   int for my type
                        second) {
  create the function
                           int temp;
                           temp = first;
                           first = second;
                           second = temp;
```

MICHIGAN STATE

generic function

By writing the function as a template, we can write a *generic function*:

• a function which, even in C++ (which is type crazy), is generic **for all types**.

Remember: a template is a pattern to make a function. *A template is not a function*

MICHIGAN STATE UNIVERSITY

force the type

Typically the compiler deduces the type for substitution in the template from the provided arguments

You can force (though you must be careful) the type used, but it has to work with the args and the created function

MICHIGAN STATE

Ex 6.8, force the template type

Invocation

```
double result;
long i=1, j=2;
result = swap double (i,j);
```

template type directly indicated

Will see this again and again. We specify in the invocation the type we want used in the template