Advanced Software Techniques

Pointers to Functions

In DSAD,
we've seen some
function names
used as
function arguments

e.g. an overloaded comparison function

for a class

stored in a set

e.g. for_each(vs.begin(), vs.end(), display_backwards);

The name of a function

is the same

as a pointer to it

Yes,

just like an array

That's easy!

Specifying the type of the parameter is a bit more clunky

To specify
a function pointer
as a parameter:

1. Return type
2. Asterisk before name
3. Parameter data types (within brackets)

e.g.

```
int (*compare)(char *, char *)
```

This is

"a pointer to a function

that takes two char * as parameters

and

returns an int"

qsort uses this

```
void qsort (void *base, unsigned
  int num, unsigned int size,
int (*compare)(const void *,
      const void*));
```

This sorts an array of something using the quicksort algorithm void *base: pointer to start of array

unsigned int num:

how many elements

unsigned int size:

how big is one element

int (*compare)(const void *,
const void*):

pointer to comparison function

e.g.

```
qsort(arrayOfGlorts,
kHowManyGlorts,
kHowBigIsAGlort,
compareGlort);
```

compareGlort must fit the description in the prototype

Example

qsort.c is online

Highlights of qsort.c

```
int a[20] = {
19, 99, 5, 10, 7,
33, 22, 57, 50, 48,
3, 77, 66, 13, 18,
9, 10, 59, 38, 28
};
```

```
int up(const void *x, const void *y);

int down(const void *x, const void *y);
```

```
qsort (a, 20, sizeof (a[0]), up);
...
qsort (a, 20, sizeof (a[0]), down);
```

Another use of Function Pointers

Jump Tables are

arrays of function pointers

Use the index
into the array
to determine
which function to call

Sample declaration:

```
int (*ptrsToFn[5])(int)
=
{func1, func2, func3, func4,
func5};
```

And to call it, just use the name as normal:

```
retValue = ptrsToFn[2](3);
```

Example

jumptable.c is online

Highlights of jumptable.c

```
int func_5 (int);

int (*ptrs_to_fn[5])(int) = {
func_1,
func_2,
func_3,
func_4,
func_5
};
```

```
y = ptrs_to_fn[x] (x + 5);
```

Why?

You'll need this for
RTOS next semester
and
it will definitely help in understanding
C# delegates next semester