INF-1100
Pointers II

Åge Kvalnes

University of Tromsø, Norway

October 17, 2014

Summary

- x[0] equals the value of entry 0 in array **x**.
- &x[0] equals the memory address of entry 0 in array **x**.
- int *p **p** is a pointer to an integer/**p** holds the memory address of an integer.
- z.y equals the value of the field **y** in the composite data structure **z**.
- &z.y equals the memory address of the field **y** in the composite data structure **z**.
- p→y equals the value of the field **y** in the composite datastructure pointed to by **p**.
- A pointer is assumed to contain the memory address of entry 0 in an array.

The duality of pointers and arrays

```
int myfunction(int p[])
{
  p[0] = 42;
}
int myfunction2(int *p)
{
  p[0] = 42;
}

int main(int argc, char **argv)
{
  int a[10];
  myfunction(&a[0]);
  myfunction(a);
  myfunction2(&a[0]);
  myfunction2(a);
}
```

The *

```
int myfunction(int *p)
{
  *p = 42;
}

int main(int argc, char **argv)
{
  int a[10];
  myfunction(a);
}
```

* in front of a pointer refers to the value pointed to.

Pointer arithmetic

```
int myfunction(int *p)
{
  *p = 42;
  p = p + 1;
  *p = 43;
}

int main(int argc, char **argv)
{
  int a[10];
  myfunction(a);
  // a[0] = 42, a[1] = 43
}
```

Incrementing p makes p point to the next array entry.

Pointer arithmetic

```
typedef struct mystruct mystruct_t;
struct mystruct {
  int number;
};

int myfunction(mystruct_t *p)
{
    p->number = 42;
    p = p + 1;
    p->number = 43;
}

int main(void)
{
  mystruct_t a[10];
  myfunction(a);
  // a[0].number = 42, a[1].number = 43
}
```

Initializing arrays

```
int a[] = {0, 1, 1, 2, 3, 5};
char s[] = "foobar";
```

Place the initializing values inside the curly braces or, for strings, quotes.

```
mystruct_t a[] = {{0}, {1}, {1}, {2}, {3}, {5}};
```

Composite data structures can be initialized similarly.

```
int *a = {0, 1, 1, 2, 3, 5};
```

WILL NOT WORK. a IS A POINTER IN THIS CONTEXT.

Initializing arrays by referring to field names

```
typedef struct mystruct mystruct_t;
struct mystruct {
  int foo;
  int bar;
};

mystruct_t a[] = {
  {.foo=0, .bar=1},
  {.foo=1, .bar=2}
  {.foo=3, .bar=5}
  {.foo=8, .bar=13}
};
```

Field names are prepended with a **dot** and separated by a **comma**.

Drawing circles

Problem: Given a description of a set of circles (origin coordinate, radius), draw the circles on the screen.

Algorithm:

1. For each circle, calculate the coordinates of points on the circle arc and draw those points on the screen.