CS 241
Data Organization
Quiz 6

Brooke Chenoweth

University of New Mexico

Fall 2014

Question 1: Structures and Functions

```
struct Point {int x; int y;};

struct Point incPoint(struct Point p)
{
   p.x++;
   p.y++;
   return p;
}

void main(void)
{
   struct Point p1 = {12, 3};
   struct Point p2 = incPoint(p1);
   printf("p1=(%d, %d) p2=(%d, %d)\n",
      p1.x, p1.y, p2.x, p2.y);
}
```

A `p1=(12, 3) p2=(13, 4)`

B `p1=(12, 3) p2=(12, 3)`

C `p1=(13, 4) p2=(13, 4)`

D `p1=(13, 4) p2=(12, 3)`

E The value returned into p2 was stored on the stack in `incPoint`. Therefore, the value in p2 is unpredictable!

Question 1: Structures and Functions

```
struct Point {int x; int y;};

struct Point incPoint(struct Point p)
{
  p.x++;
  p.y++;
  return p;
}

void main(void)
{
  struct Point p1 = {12, 3};
  struct Point p2 = incPoint(p1);
  printf("p1=(%d, %d) p2=(%d, %d)\n",
     p1.x, p1.y, p2.x, p2.y);
}
```

A `p1=(12, 3) p2=(13, 4)`

B `p1=(12, 3) p2=(12, 3)`

C `p1=(13, 4) p2=(13, 4)`

D `p1=(13, 4) p2=(12, 3)`

E The value returned into `p2` was stored on the stack in `incPoint`. Therefore, the value in `p2` is unpredictable!

Question 2: Pointers to Structures

```
struct Point {int x; int y;};

void incrementPoint(struct Point *p)
{
   (*p).x += 2;
   p->y   += 2;
}

void main(void)
{
   struct Point p1 = {7, 7};
   incrementPoint(&p1);
   printf("p1=(%d, %d)\n", p1.x, p1.y);
}
```

A p1=(7, 7)

B p1=(7, 9)

C p1=(9, 9)

D p1=(9, 7)

E p1 = 14

Question 2: Pointers to Structures

```
struct Point {int x; int y;};

void incrementPoint(struct Point *p)
{
   (*p).x += 2;
   p->y   += 2;
}

void main(void)
{
   struct Point p1 = {7, 7};
   incrementPoint(&p1);
   printf("p1=(%d, %d)\n", p1.x, p1.y);
}
```

A p1=(7, 7)

B p1=(7, 9)

C p1=(9, 9)

D p1=(9, 7)

E p1 = 14

Question 3: Pointers to Structures

```
#include <stdio.h>
#include <math.h>
struct Point {double x; double y;};
void foo(struct Point *p)
{
   double d = sqrt((p->x)*(p->x)
                   + (p->y)*(p->y));
   p->x    /= d;
   p->y    /= d;
}
void main(void)
{
   struct Point p1 = {3, 4};
   foo(&p1);
   printf("p1=(%5.2f, %5.2f)\n", p1.x, p1.y);
}
```

A p1=(1.67, 1.25)

B p1=(0.60, 0.80)

C p1=(3.00, 4.00)

D p1=(3, 4)

E p1=(0.12, 0.16)

Question 3: Pointers to Structures

```
#include <stdio.h>
#include <math.h>
struct Point {double x; double y;};
void foo(struct Point *p)
{
   double d = sqrt((p->x)*(p->x)
                   + (p->y)*(p->y));
   p->x    /= d;
   p->y    /= d;
}
void main(void)
{
   struct Point p1 = {3, 4};
   foo(&p1);
   printf("p1=(%5.2f, %5.2f)\n", p1.x, p1.y);
}
```

A p1=(1.67, 1.25)

B p1=(0.60, 0.80)

C p1=(3.00, 4.00)

D p1=(3, 4)

E p1=(0.12, 0.16)

Question 4: Pointer and Index

```
void main(void)
{
   char data[] = "Warcraft";
   data[7] = '+';
   char *linePt = &data[4];
   *linePt = '*';
   printf("[%s], [%s]\n", data, linePt);
}
```

A [Warcraft], [Warcraft]

B [Warcraf+], [Warc*aft]

C [Warc*aft], [Warcraf+]

D [Warc*aft], [raf+]

E [Warc*af+], [*af+]

Question 4: Pointer and Index

```
void main(void)
{
   char data[] = "Warcraft";
   data[7] = '+';
   char *linePt = &data[4];
   *linePt = '*';
   printf("[%s], [%s]\n", data, linePt);
}
```

A [Warcraft], [Warcraft]

B [Warcraf+], [Warc*aft]

C [Warc*aft], [Warcraf+]

D [Warc*aft], [raf+]

E [Warc*af+], [*af+]