

CS 241

Data Organization

Quiz 5

March 8, 2018

Question 1: Structures and Functions

```
struct Point {int x; int y;};
struct Point incPoint(struct Point p)
{
    p.x++;
    p.y++;
    return p;
}
int main(void)
{
    struct Point p1 = {4, 3};
    struct Point p2 = incPoint(p1);
    printf("p1=(\%d, \%d)
           p2=(\%d, \%d)\n",
           p1.x, p1.y, p2.x, p2.y);
}
```

A p1=(4, 3) p2=(5, 4)

B p1=(4, 3) p2=(4, 3)

C p1=(5, 4) p2=(4, 3)

D p1=(5, 4) p2=(5, 4)

E 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;
}
int main(void)
{
    struct Point p1 = {4, 3};
    struct Point p2 = incPoint(p1);
    printf("p1=(\%d, \%d)
           p2=(\%d, \%d)\n",
           p1.x, p1.y, p2.x, p2.y);
}
```

A p1=(4, 3) p2=(5, 4)

B p1=(4, 3) p2=(4, 3)

C p1=(5, 4) p2=(4, 3)

D p1=(5, 4) p2=(5, 4)

E The value in p2 is unpredictable.

Question 2: Pointer and Index

```
int main(void)
{
    char data[] = "computer science";
    data[12] = '*';
    char *linePt = &data[4];
    *linePt = '+';
    printf("[\\%s], [\\%s]\\n", data, linePt);
    return 0;
}
```

- A [computer science], [computer science]
- B [comp+ter sci*nice], [comp+ter sci*nice]
- C [comp+ter science], [computer sci*nice]
- D [comp+ter sci*nice], [uter sci*nice]
- E [comp+ter sci*nice], [+ter sci*nice]

Question 2: Pointer and Index

```
int main(void)
{
    char data[] = "computer science";
    data[12] = '*';
    char *linePt = &data[4];
    *linePt = '+';
    printf("[\\%s], [\\%s]\\n", data, linePt);
    return 0;
}
```

- A [computer science], [computer science]
- B [comp+ter sci*nice], [comp+ter sci*nice]
- C [comp+ter science], [computer sci*nice]
- D [comp+ter sci*nice], [uter sci*nice]
- E [comp+ter sci*nice], [+ter sci*nice]

Question 3: Pointers to Structures

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

```
void incrementPoint(struct Point *p)
```

```
{
    (*p).x += 2;
    p->y    += 2;
}
```

```
int main(void)
```

```
{
    struct Point p1 = {1, 1};
    incrementPoint(&p1);
    printf("p1=(%d, %d)\n", p1.x, p1.y);
    return 0;
}
```

A p1=(3, 3)

B p1=(1, 3)

C p1=(1, 1)

D p1=(3, 1)

E p1 = 2

Question 3: Pointers to Structures

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

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

```
int main(void)
{
    struct Point p1 = {1, 1};
    incrementPoint (&p1);
    printf("p1=(%d, %d)\n", p1.x, p1.y);
    return 0;
}
```

A p1=(3, 3)

B p1=(1, 3)

C p1=(1, 1)

D p1=(3, 1)

E p1 = 2

Question 4: 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 = {5, 4};
    foo(&p1);
    printf("p1=(%5.2f, %5.2f)\n", p1.x, p1.y);
}
```

A p1=(5, 4)

B p1=(0.78, 0.62)

C p1=(5.00, 4.00)

D p1=(0.60, 0.80)

E p1=(0.12, 0.98)

Question 4: 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 = {5, 4};
    foo(&p1);
    printf("p1=(%5.2f, %5.2f)\n", p1.x, p1.y);
}
```

A p1=(5, 4)

B p1=(0.78, 0.62)

C p1=(5.00, 4.00)

D p1=(0.60, 0.80)

E p1=(0.12, 0.98)