

# Data Types

CS 3410: Computer System Organization and Programming





# Type Aliases

```
#include <stdio.h>

typedef int number;

void main() {
    number x = 3410;
    int y = x / 2;
    printf("%d %d!\n", x, y);
}
```



# Structures

```
#include <stdio.h>
```

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

```
void print_point(struct point p) {  
    printf("(%d, %d)\n", p.x, p.y);  
}
```

```
void main() {  
    struct point location = {4, 10};  
    location.y = 2;  
    print_point(location);  
}
```

Supply all the fields, in order, in the curly braces of the initializer.



# Short Names for Structs

Give struct a short name using typedef

```
#include <stdio.h>
```

```
typedef point {  
    int x;  
    int y;  
} point_t;
```

Use <name>\_t for custom type names

```
void print_point(point_t p) {  
    printf("(%d, %d)\n", p.x, p.y);  
}
```

```
void main() {  
    point_t location = {4, 10};  
    location.y = 2;  
    print_point(location);  
}
```



# Enumerations

```
#include <stdio.h>
```

```
typedef enum {  
    SPRING,  
    SUMMER,  
    AUTUMN,  
    WINTER,  
} season_t;
```

```
int main() {  
    season_t now = WINTER;  
    season_t next = SPRING;  
    printf("%d %d\n", now, next);  
    return 0;  
}
```

Useful to make code more readable to use a name where you may have used an integer

