

# CSC 211: Computer Programming

## Structs

Michael Conti

Department of Computer Science and Statistics  
University of Rhode Island

Summer 2025



## Structures

- Definition is generally outside any function
  - new 'data type' will be available to all code that follows
- Structures can be declared in the same way as basic data types
- Can also use { } notation for initialization
- Use the **dot operator** for accessing data members

2

## Structures

```
struct structureName {  
    member1;  
    member2;  
    member3;  
    .  
    .  
    .  
    memberN;  
};
```

Structures in C++ are user defined data types which are used to store multiple items (members) of possibly different data types

3

## Example

```
// defining the struct  
struct Point {  
    int x;  
    int y;  
};  
  
int main() {  
    // creating a variable  
    struct Point p1;  
}
```

4

## Initializing ...

```
// defining the struct
struct Point {
    int x;
    int y;
};

int main() {
    // initializing (follows order)
    struct Point p1 = { 10, 20 };
}
```

5

## The dot operator

```
#include <iostream>

struct Point {
    int x;
    int y;
};

int main() {
    struct Point p1 = { 10, 20 };
    p1.x += 5;
    std::cout << p1.x << ' ' << p1.y << '\n';
}
```

6

## The dot operator

```
#include <iostream>

struct Point {
    int x;
    int y;
};

int main() {
    struct Point p1 = { 10, 20 };
    struct Point p2 = { 30, 40 };
    struct Point p3 = { 50, 60 };
    p1.x += 5; p2.y += 10; p3.y += 15;
}
```

7

### DISPLAY 10.2 Member Values

```
1 struct CDAccount
2 {
3     double balance;
4     double interestRate;
5     int term; //months until maturity
6 };
7 int main( )
8 {
9     CDAccount account;
10     ...
11
12
13     account.balance = 1000.00;
14
15
16     account.interestRate = 4.7;
17
18
19     account.term = 11;
20
21
22
```

State	balance	interestRate	term
Initial	?	?	?
After account.balance = 1000.00;	1000.00	?	?
After account.interestRate = 4.7;	1000.00	4.7	?
After account.term = 11;	1000.00	4.7	11

from: Problem Solving with C++, 10th Edition, Walter Savitch

8

# Array of structures

```
#include <iostream>

struct Point2D {
    double x;
    double y;
};

int main() {

    Point2D mypoint;
    Point2D myarray[5];

    mypoint.x = 10;
    mypoint.y = 20;

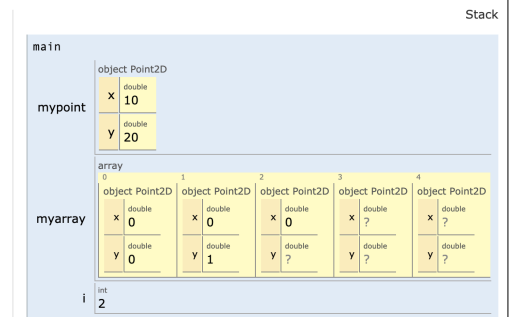
    for (int i = 0 ; i < 5 ; i++) {
        myarray[i].x = 0;
        myarray[i].y = i;
    }
}
```

9

# pythontutor.com

C++ (gcc 4.8, C++11)  
EXPERIMENTAL! known limitations

```
1 struct Point2D {
2     double x;
3     double y;
4 };
5
6 int main() {
7     struct Point2D mypoint;
8     struct Point2D myarray[5];
9
10    mypoint.x = 10;
11    mypoint.y = 20;
12
13    for (int i = 0 ; i < 5 ; i++) {
14        myarray[i].x = 0;
15        myarray[i].y = i;
16    }
17
18 }
```



10

## Arrays and Structures

- When using arrays as structs member, the index goes at the end
  - ✓ student.grades[i]
- When using structs as arrays elements, the index goes after the struct name
  - ✓ students[i].finalGrade

11

## Functions

```
// defining the struct
struct Point {
    int x;
    int y;
};

void distance(Point P1, Point P2);
```

12

## Passing structures to functions

- A struct can be passed as a parameter either by value or by reference

```
void printPoint(Point &somePoint){
    std::cout << somePoint.x;
    std::cout << somePoint.y;
};
```

- A function can return a value of type struct

```
Point incrementPoint(Point somePoint){
    somePoint.x+=1;
    somePoint.y+=1;
    return somePoint;
};
```

13

## Passing structures to functions

DISPLAY 10.1 A Structure Definition ..

```
1 //Program to demonstrate the CDAccount structure type.
2 #include <iostream>
3 using namespace std;
4 //Structure for a bank certificate of deposit:
5 struct CDAccount
6 {
7     double balance;
8     double interestRate;
9     int term; //months until maturity
10 };
11
12
13 void getData(CDAccount& theAccount);
14 //Postcondition: theAccount.balance and theAccount.interestRate
15 //have been given values that the user entered at the keyboard.
16
17
18 int main( )
19 {
20     CDAccount account;
21     getData(account);
22
23     double rateFraction, interest;
24     rateFraction = account.interestRate / 100.0;
25     interest = account.balance * rateFraction * (account.term / 12.0);
26     account.balance = account.balance + interest;
27
28     cout.setf(ios::fixed);
29     cout.setf(ios::showpoint);
30     cout.precision(2);
31     cout << "When your CD matures in "
32           << account.term << " months,\n"
33           << "it will have a balance of $"
34           << account.balance << endl;
35     return 0;
36 }
```

```
38 //Uses iostream:
39 void getData(CDAccount& theAccount)
40 {
41     cout << "Enter account balance: $";
42     cin >> theAccount.balance;
43     cout << "Enter account interest rate: ";
44     cin >> theAccount.interestRate;
45     cout << "Enter the number of months until maturity\n"
46           << "(must be 12 or fewer months): ";
47     cin >> theAccount.term;
48 }
```

from: Problem Solving with C++, 10th Edition, Walter Savitch

14

## Be careful of same member names

```
// defining the struct
struct Point {
    int x;
    int y;
};

struct Character {
    int x;
    int y;
    std::string name;
};
```

Compiler can keep track but it's harder for humans

15

## Structs and Pointers

```
struct Books {
    std::string title;
    std::string author;
    std::string subject;
    int book_id;
};
```

16

## Pointers and Structs

- You can define pointers to structures in very similar way as you define pointer to any other variable

```
struct Books *struct_pointer;  
struct Books Book1;
```

- Now, you can store the address of a structure variable in the above defined pointer variable.

```
struct_pointer = &Book1;
```

17

## Structs and Pointers

```
void printBook( struct Books *book ) {  
    std::cout << "Book title : " << (*book).title;  
    std::cout << "Book author : " << (*book).author;  
    std::cout << "Book subject : " << (*book).subject;  
    std::cout << "Book id : " << (*book).book_id;  
}  
  
struct Books {  
    std::string title;  
    std::string author;  
    std::string subject;  
    int book_id;  
};  
  
int main(){  
    struct Books Book1;  
  
    Book1.title = "Learn C++ Programming"  
    Book1.author = "Chand Miyan"  
    Book1.subject = "Computer Science"  
  
    printBook( &Book1 );  
}
```

18

## Example

- Write a Student struct that contains
  - ✓ Name
  - ✓ StudentID
  - ✓ Major
- Implement functions:
  - ✓ void buildStudent(Student &someStudent)
    - Initialize member variables of student Struct
  - ✓ void changeMajor(Student &someStudent);
    - Change the major of a student structure
  - ✓ void printStudent(Student &someStudent);
    - Prints out all member variables of student structure

19