

# CSC 211: Object Oriented Programming

## Structs

Michael Conti

Department of Computer Science and Statistics  
University of Rhode Island

Spring 2020



Original design and development by Dr. Marco Alvarez

## 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

2

## 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

3

### 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
```

balance	interestRate	term	
?	?	?	account
1000.00	?	?	
?	?	?	
1000.00	4.7	?	account
1000.00	4.7	11	

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

4

## Example

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

int main() {
    // creating a variable
    struct Point p1;
}
```

5

## Initializing ...

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

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

6

## 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';
}
```

7

## 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;
    }
}
```

8

# 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 }
```

# 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 }
```

Stack

