

## 7.1 Grouping data: struct

Sometimes two data items are really aspects of the same data. For example, time might be recorded in hours and minutes, as in 4 hours and 23 minutes. Or a point on a plot might be recorded as  $x = 5$ ,  $y = 7$ . Storing such data in separate variables, such as `runTimeHours` and `runTimeMinutes`, is not as clear as grouping that data into a single variable, like `runTime`, which might have subitems `runTime.hourValue` and `runTime.minuteValue`.

PARTICIPATION  
ACTIVITY

7.1.1: Naturally grouped data.

1) Select the pair forming part of a person's height (in U.S. units)

☐ Feet and inches

☐ Inches and salary

☐ Pounds and ounces

2) Select the group of items indicating the change provided to a person who pays for a meal.

☐ Ounce, gill, pint, quart, and gallon

☐ Mile, furlong, yard, feet, and inches

☐ Dollars, quarters, dimes, nickels, and pennies

The **struct** construct defines a new type, which can be used to declare a variable with subitems. The following animation illustrates.

PARTICIPATION  
ACTIVITY

7.1.2: A struct enables creating a variable with data members.

```
struct TimeHrMin {
    int hourValue;
    int minuteValue;
};

...

TimeHrMin runTime1;
TimeHrMin runTime2;
TimeHrMin runTime3;
```

96	5	hourValue	runTime1
97	46	minuteValue	
98	?	hourValue	runTime2
99	?	minuteValue	
100	5	hourValue	runTime3
101	?	minuteValue	
102			

```
runTime1.hourValue = 5;
runTime1.minuteValue = 46;
runTime3.hourValue = runTime1.hourValue;
```

## Animation content:

Code snippet is as follows:

```
struct TimeHrMin {
    int hourValue;
    int minuteValue;
};
```

...

```
TimeHrMin runTime1;
TimeHrMin runTime2;
TimeHrMin runTime3;
```

```
runTime1.hourValue = 5;
runTime1.minuteValue = 46;
runTime3.hourValue = runTime1.hourValue;
```

Final memory contents is as follows:

```
96 (runTime1's hourValue): 5
97 (runTime1's hourValue): 46
98 (runTime2's hourValue): ?
99 (runTime2's hourValue): ?
100 (runTime3's hourValue): 5
101 (runTime3's hourValue): ?
102: empty
```

## Animation captions:

1. The struct construct just declares new type; no memory is allocated.
2. Variable definitions allocate memory for each object's member.
3. Accesses refer to an object member's memory location.

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

The programmer uses a struct to define and use a new type as follows.

### Construct 7.1.1: Defining and using a new struct type.

```
struct StructTypeName
{
    type item1;
    type item2;
    ...
    type itemN;
};

...
StructTypeName myVar;

myVar.item1 = ...
```

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

Each `type` may be any type like `int` or `char`. Each struct subitem is called a **data member**. For a declared variable, each struct data member can be accessed using `"."`, known as a **member access** operator, sometimes called **dot notation**.

Assigning a variable of a struct type to another such variable automatically assigns each corresponding data member, as shown below.

PARTICIPATION  
ACTIVITY

7.1.3: Assigning a struct type.

```
struct TimeHrMin {
    int hourValue;
    int minuteValue;
};

...

TimeHrMin runTime1;
TimeHrMin runTime2;
TimeHrMin runTime3;

runTime1.hourValue = 5;
runTime1.minuteValue = 46;
runTime2 = runTime1;
```

96	5	hourValue	runTime1
97	46	minuteValue	
98	5	hourValue	runTime2
99	46	minuteValue	
100	?	hourValue	runTime3
101	?	minuteValue	
102			

Animation content:

Code snippet is as follows:

```
struct TimeHrMin {
    int hourValue;
    int minuteValue;
};

...
```

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

```
TimeHrMin runTime1;
TimeHrMin runTime2;
TimeHrMin runTime3;
```

```
runTime1.hourValue = 5;
runTime1.minuteValue = 46;
runTime2 = runTime1;
```

Final memory contents is as follows:

```
96 (runTime1's hourValue): 5
97 (runTime1's hourValue): 46
98 (runTime2's hourValue): ?
99 (runTime2's hourValue): ?
100 (runTime3's hourValue): 5
101 (runTime3's hourValue): ?
102: empty
```

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

### Animation captions:

1. Assigning a variable of a struct type to another such variable automatically assigns each corresponding data member.

#### PARTICIPATION ACTIVITY

#### 7.1.4: The struct construct.

- 1) A struct definition for CartesianPoint has subitems int x and int y. How many int locations in memory does the struct definition allocate?

Check

Show answer

- 2) If struct definition CartesianPoint has subitems int x and int y, how many total int locations in memory are allocated for these variable declarations?

```
int myNum;
CartesianPoint myPoint1;
CartesianPoint myPoint2;
```

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

**Check****Show answer**

- 3) Given time1 is of type TimeHrMin defined earlier. What is the value of variable min after the following statements?

```
time1.hourValue = 5;  
time1.minuteValue = 4;  
min = (60 * time1.hourValue)  
+ time1.minuteValue;
```

**Check****Show answer**

- 4) Write a statement to assign 12 to the hourValue data member of TimeHrMin variable time1.

**Check****Show answer**

- 5) Write a statement that assigns the value of the hourValue data member of time1 into the hourValue data member of time2.

**Check****Show answer**

- 6) Write a single statement that assigns the values of all data members of time1 to the corresponding data members of time2.

**Check****Show answer**

- 7) Declare a variable person1 of type Person, where Person is already defined as a struct type.

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

Check

Show answer

CHALLENGE  
ACTIVITY

7.1.1: Enter the output using struct.

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

602398.4584902.qx3zqy7

Start

Type the program's output

```
#include <iostream>
using namespace std;

struct Height {
    int feet;
    int inches;
};

int main() {
    Height annHeight;

    annHeight.feet = 4;
    annHeight.inches = 10;

    cout << "Ann: " << annHeight.feet << "ft " << annHeight.inches << endl;

    return 0;
}
```

Ann: 4ft 10in

1

2

3

4

Check

Next

CHALLENGE  
ACTIVITY

7.1.2: Declaring a struct.

Full screen

602398.4584902.qx3zqy7

Organize the lines of code to define a struct named PatientData that contains two integer data member: and weightPounds.

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

Ex. If the input is 63 115 then the output is:

Patient data: 63 in, 115 lbs

Note: This activity includes distractors. Not all lines of code on the left will be used in the final solution.

How to use this tool ▼

## Unused

```

struct patientData {
struct PatientData {
int weightPounds;
int heightInches;
}
};

```

## main.cpp

```

#include <iostream>
using namespace std;

int main() {
    PatientData lunaLovegood;

    cin >> lunaLovegood.heightInch
    cin >> lunaLovegood.weightPoun

    cout << "Patient data: "
        << lunaLovegood.heightInc
        << lunaLovegood.weightPou

    return 0;
}

```

Check

CHALLENGE  
ACTIVITY

7.1.3: Accessing a struct's data members.

Write a statement to print the data members of InventoryTag. End with newline. Ex: if itemID is 314 and quantityRemaining is 500, print:

Inventory ID: 314, Qty: 500

[Learn how our autograder works](#)

602398.4584902.qx3zqy7

```

1 #include <iostream>
2 using namespace std;
3
4 struct InventoryTag {
5     int itemID;
6     int quantityRemaining;
7 };
8
9 int main() {
10     InventoryTag redSweater;
11
12     cin >> redSweater.itemID;
13     cin >> redSweater.quantityRemaining;

```

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

14

15

16

/\* Your solution goes here \*/

Run

## 7.2 Structs and functions

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

The struct construct's power is evident when used with functions. A struct can be used to return multiple values. Although ConvHrMin() has two output values, the struct type allows the function to return a single item, avoiding a less-clear approach using two pass by reference parameters.

PARTICIPATION  
ACTIVITY

7.2.1: Using a struct that is returned from a function; the struct's data members are copied upon return.

```
#include <iostream>
using namespace std;

struct TimeHrMin {
    int hourValue;
    int minuteValue;
};

TimeHrMin ConvHrMin(int totalTime) {
    TimeHrMin timeStruct;

    timeStruct.hourValue = totalTime / 60;
    timeStruct.minuteValue = totalTime % 60;

    return timeStruct;
}

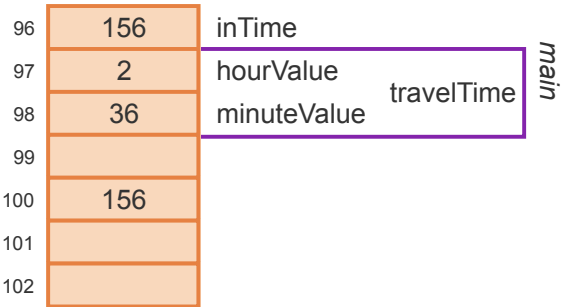
int main() {
    int inTime;
    TimeHrMin travelTime;

    cout << "Enter total minutes: ";
    cin >> inTime;

    travelTime = ConvHrMin(inTime);

    cout << "Equals: ";
    cout << travelTime.hourValue << " hrs ";
    cout << travelTime.minuteValue << " mins";

    return 0;
}
```



Enter total minutes: 156  
Equals: 2 hrs 36 mins

### Animation content:

Code snippet is as follows:  
#include



```
using namespace std;
```

```
struct TimeHrMin {
    int hourValue;
    int minuteValue;
};
```

```
TimeHrMin ConvHrMin(int totalTime) {
    TimeHrMin timeStruct;
```

```
    timeStruct.hourValue = totalTime / 60;
    timeStruct.minuteValue = totalTime % 60;
```

```
    return timeStruct;
}
```

```
int main() {
    int inTime;
    TimeHrMin travelTime;
```

```
    cout << "Enter total minutes: ";
    cin >> inTime;
```

```
    travelTime = ConvHrMin(inTime);
```

```
    cout << "Equals: ";
    cout << travelTime.hourValue << " hrs ";
    cout << travelTime.minuteValue << " mins";
```

```
    return 0;
}
```

Final memory contents is as follows:

```
96 (main's inTime): 156
97 (main's travelTime hourValue): 2
98 (main's's travelTime hourValue): 36
99: empty
100 (ConvHrMin's totalTime): 156
101 (ConvHrMin's timeStruct hourValue): 2
102 (ConvHrMin's timeStruct minuteValue): 36
```

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

### Animation captions:

1. The program prompts a user to enter travel time in minutes, then calls the ConvHrMin function to convert travel time to hours and minutes.

2. Upon return, timeStruct's data members are copied to main's travelTime variable.
3. Returning a struct type allows the ConvHrMin function to return a single item, avoiding a less-clear approach of using two pass-by-reference parameters.

### zyDE 7.2.1: Monetary change program.

Complete the program to compute monetary change, using the largest coins possible.

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

Load default template...

```

1 #include <iostream>
2 using namespace std;
3
4 struct MonetaryChange {
5     int quarters;
6     // FIXME: Finish data members
7 };
8
9 MonetaryChange ComputeChange(int amount) {
10     MonetaryChange change;
11
12     // FIXME: Finish function body
13     change.quarters = 0; // Example line
14
15     return change;
16 }
```

119

Run

#### PARTICIPATION ACTIVITY

#### 7.2.2: Functions returning struct values.

- 1) Complete the function definition for a function ComputeLocation that returns a struct of type GPSPosition.

```
(double latitude, double longitude) {
```

...

```
}
```

Check

Show answer

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

- 2) Complete the function to return the calculated elapsed time, which gets stored in `elapsedTime`.

```
TimeEntry CalcElapsedTime(int
startSecs, int endSecs) {

    TimeEntry elapsedTime;

    ...

    elapsedTime.totalSecs =
endSecs - startSecs;

    elapsedTime.hours =
(endSecs - startSecs) / 3600;

    ...

    
;
}
```

Check

Show answer

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

Likewise, a variable of a struct type can be a function parameter. And just like other types, a pass by value parameter would copy the item, while a pass by reference parameter would not.

#### PARTICIPATION ACTIVITY

#### 7.2.3: Functions with struct parameters.

- 1) Complete the function definition for a function `CalcSpeed` that returns a double value and has two struct type parameters `startLoc` and `endLoc` (in that order) of type `GPSPosition`.

```
double CalcSpeed(

){
    ...
}
```

Check

Show answer

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

- 2) Complete the following statement to calculate the speed between a `startLoc` of `gpsPos1` and an `endLoc`

of gpsPos2 by making a call to the CalcSpeed function.

```
double vehicleSpeed;
GPSPosition gpsPos1;
GPSPosition gpsPos2;
...
vehicleSpeed =

;
...
```

Check

Show answer

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024

### CHALLENGE ACTIVITY

7.2.1: Enter the output of the struct and function.

602398.4584902.qx3zqy7

Start

Type the program's output

```
#include <iostream>
using namespace std;

struct Home {
    int numBathrooms;
    int numFloors;
    int numPeople;
    int numRooms;
};

Home InitHome() {
    Home tempHome;

    tempHome.numBathrooms = 4;
    tempHome.numFloors = 3;
    tempHome.numPeople = 8;
    tempHome.numRooms = 6;

    return tempHome;
}

int main() {
    Home myHome = InitHome();

    cout << myHome.numFloors << " Floors" << endl;
    cout << myHome.numPeople << " People" << endl;

    return 0;
}
```

3 Floors  
8 People

1

2

3

Check

Next

CHALLENGE  
ACTIVITY

## 7.2.2: Structs and functions.



602398.4584902.qx3zqy7

Start

Write a statement that calls the function `AddToStock` with parameters `mugInfo` and `addQty`.  
Assign `mugInfo` with the value returned.

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct ProductInfo {
6     string itemName;
7     int itemQty;
8 };
9
10 ProductInfo AddToStock(ProductInfo productToStock, int increaseValue) {
11     productToStock.itemQty = productToStock.itemQty + increaseValue;
12
13     return productToStock;
14 }
15
16 int main() {
17     ProductInfo mugInfo;
18     int addQty;
19
20     cin >> mugInfo.itemName >> mugInfo.itemQty;
21     cin >> addQty;
22
23     /* Your code goes here */
24
25     cout << "Name: " << mugInfo.itemName << ", stock: " << mugInfo.itemQty;
26
27     return 0;
28 }
```

1

2

3

Check

Next

©zyBooks 09/29/24 11:13 2292451  
Diwas Pandit  
TXSTATECS1428Vargas-GarzonFall2024

## 7.3 LAB: Classic Cars



This section's content is not available for print.

## 7.4 LAB: Soccer team roster (Arrays)

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024



This section's content is not available for print.

©zyBooks 09/29/24 11:13 2292451

Diwas Pandit

TXSTATECS1428Vargas-GarzonFall2024