Lecture-3 Structures

Agenda

01. Definition
02. Initialization
03. Nested Structures
04. Enumerations
05. Tasks

01.

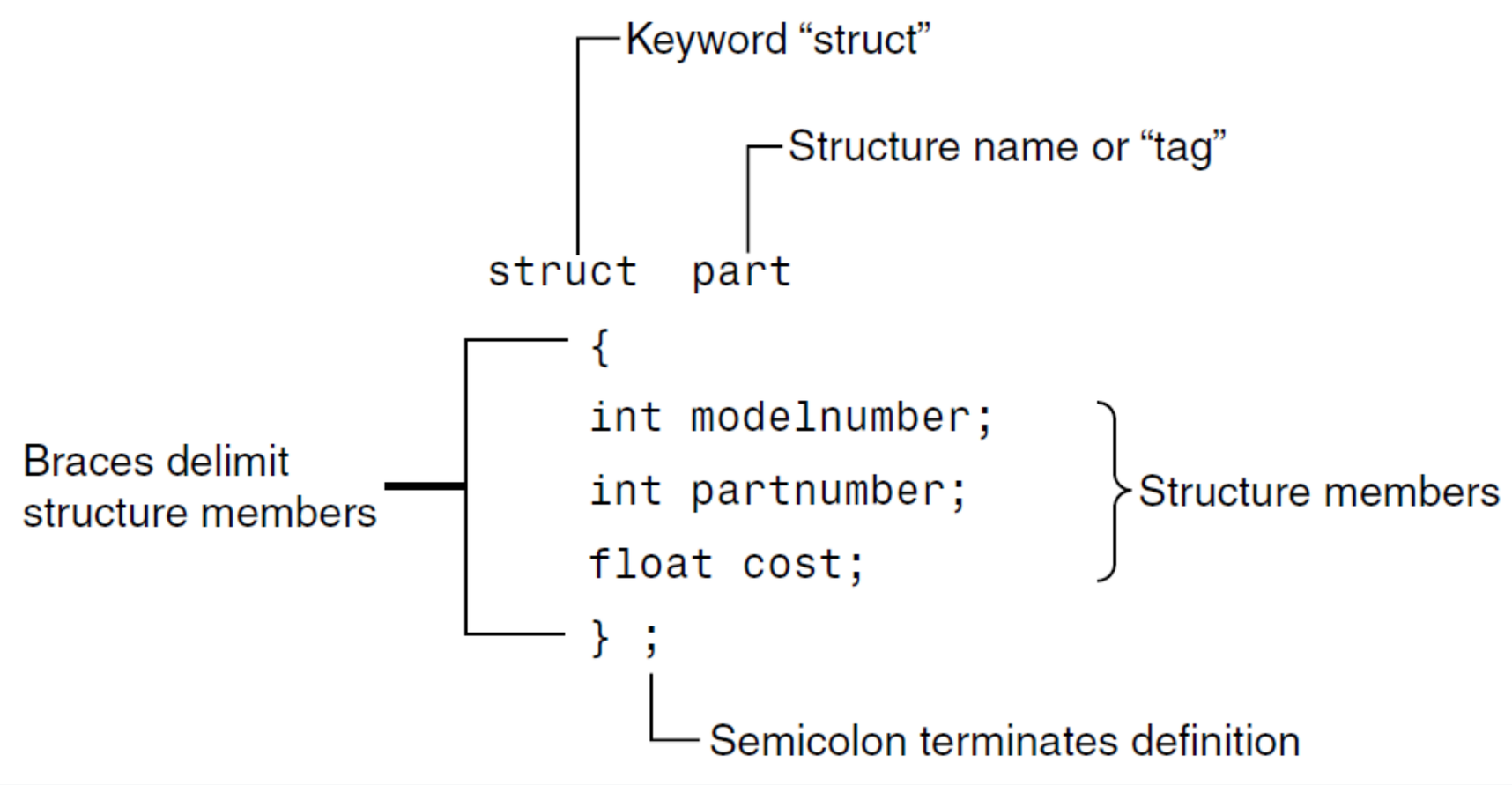
Definition

Structure

- A structure is a ***collection of variables***.
- The variables in a structure can be of ***different types.***
- The data items in a structure are called the ***members of the structure***.

Struct	Only contains variables (c). Variables can be of different data types.
Array	Only contains variables. Variables are of same data types.
Class	contains variables and functions. Variables can be of different data types.

Definition


Keyword “struct”
Structure name or “tag”
struct part
{
int modelnumber;
int partnumber;
float cost;
} ;
Braces delimit structure members
Structure members
Semicolon terminates definition

Structure Variable

```
part part1;
```

Accessing Member

```
part1.modelnumber = 6244;
```

02. Initialization

Initializing the struct variable

```
part part1 = { 6244, 373, 217.55F };
```

Definition

```
part part1 = { 6244, 373, 217.55F };
```

```
part part2;
part2 = part1;
```

Definition

```
part part1;


part1.modelnumber = 6244;
part1.partnumber = 373;
part1.cost = 217.55F;
```

Defining the Structure

A structure brings together a group of

a. items of the same data type.

b. related data items.

c. integers with user-defined names.

d. variables.

Defining the Structure

The closing brace of a structure is followed by a _____.

Defining the Structure

Write a structure specification that includes three variables—all of type `int`—called `hrs`, `mins`, and `secs`. Call this structure `time`.

Defining the Structure

True or false: A structure definition creates space in memory for a variable.

Defining the Structure

True or false: A structure definition creates space in memory for a variable.

- The structure definition serves only as a blueprint for the creation of variables of type part.
- It does not itself create any structure variables.
- It does not set aside any space in memory or even name any variables.
- This is unlike the definition of a simple variable, which does set aside memory.

Defining the Structure

When accessing a structure member, the identifier to the left of the dot operator is the name of

a. a structure member.

b. a structure tag.

c. a structure variable.

d. the keyword `struct`.

Defining the Structure

Write a statement that sets the `hrs` member of the `time2` structure variable equal to 11.

03.

Nested Structures

Structure within Structure

```
struct Distance
    {
    int feet;

     float inches;
     };
```

```
struct Room
    {
    Distance length;
    Distance width;
    };
```

Initializing the Structure

```
struct Distance
    {
    int feet;

     float inches;
     };
```

```
struct Room
   {
   Distance length;
   Distance width;
   };
```

```
Room dining;

dining.length.feet = 13;
dining.length.inches = 6.5;
dining.width.feet = 10;
dining.width.inches = 0.0;
```

Initializing the Structure

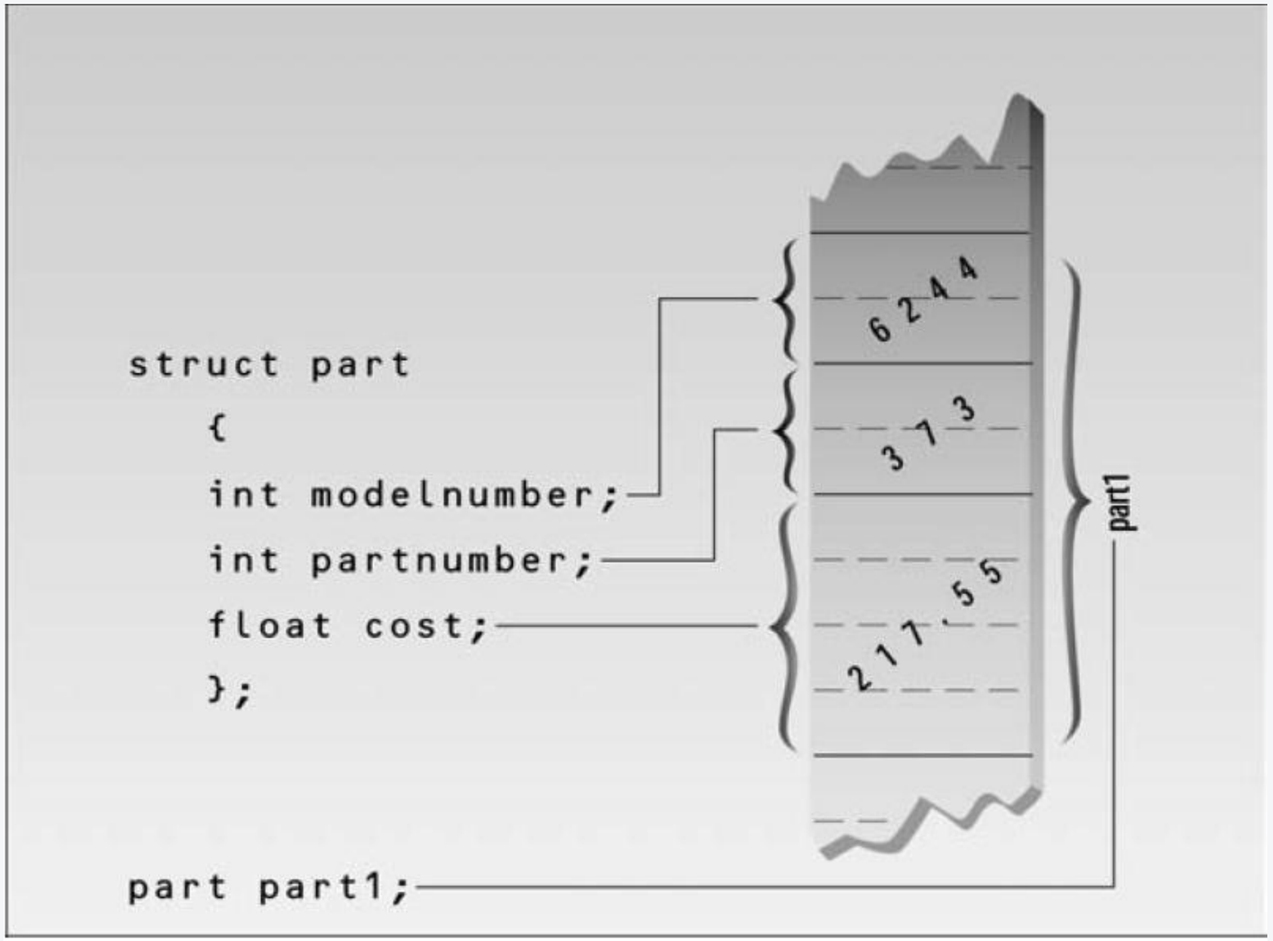
```
struct Distance
    {
    int feet;

     float inches;
     };
```

```
Room dining = { {13, 6.5}, {10, 0.0} };
```

```
struct Room
   {
   Distance length;
   Distance width;
   };
```

Structure members in memory



Structure

If you have three variables defined to be of type `struct time`, and this structure contains three `int` members, how many bytes of memory do the variables use together?

Structure

Write a definition that initializes the members of `time1`—which is a variable of type `struct time`, as defined in Question 4—to `hrs` = 11, `mins` = 10, `secs` = 59.

Structure

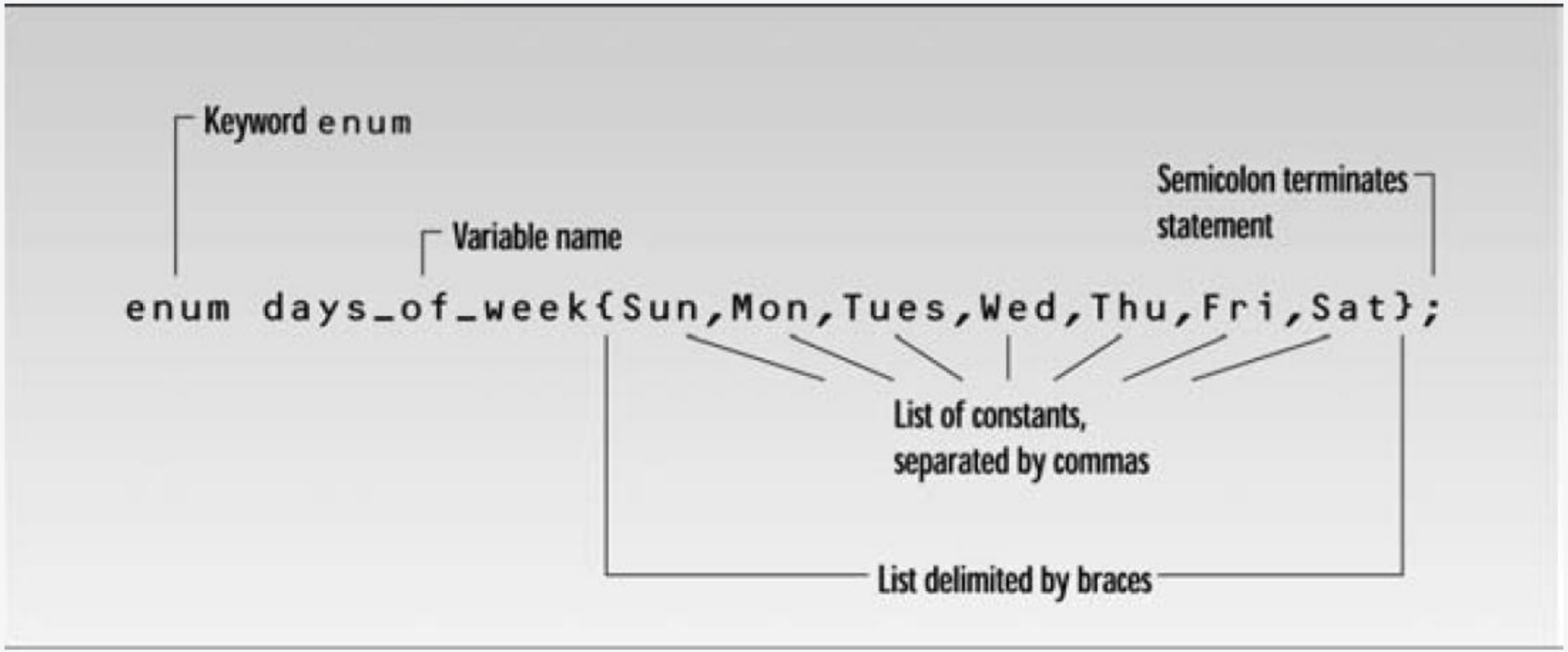
True or false: You can assign one structure variable to another, provided they are of the same type.

Structure

Write a statement that sets the variable `temp` equal to the `paw` member of the `dogs` member of the `fido` variable.

04. Enumerations

Defining the Enumeration



Enumeration

```
#include <iostream>
using namespace std;
                                  //specify enum type
enum days_of_week { Sun, Mon, Tue, Wed, Thu, Fri, Sat };

int main()
   {
   days_of_week day1, day2;    //define variables
                               //of type days_of_week

 day1 = Mon;                   //give values to
 day2 = Thu;                   //variables

 int diff = day2 - day1;       //can do integer arithmetic
 cout << "Days between = " << diff << endl;

 if(day1 < day2)               //can do comparisons
    cout << "day1 comes before day2\n";
 return 0;
 }
```

Defining the Enumeration

```
enum Suit { clubs=1, diamonds, hearts, spades };
```

Enumeration

An enumeration brings together a group of

a. items of different data types.

b. related data variables.

c. integers with user-defined names.

d. constant values.

Enumeration

Write a statement that declares an enumeration called `players` with the values B1, B2, SS, B3, RF, CF, LF, P, and C.

Enumeration

Assuming the `enum` type `players` as declared in Question 13, define two variables `joe` and `tom`, and assign them the values LF and P, respectively.

Enumeration

Assuming the statements of Questions 13 and 14, state whether each of the following statements is legal.

a. `joe = QB;`

b. `tom = SS;`

c. `LF = tom;`

d. `difference = joe - tom;`

Enumeration

The first three enumerators of an `enum` type are normally represented by the values _____, _____, and _____.

Enumeration

Write a statement that declares an enumeration called `speeds` with the enumerators `obsolete`, `single`, and `album`. Give these three names the integer values 78, 45, and 33.

Enumeration

State the reason that

```
enum isWord{ NO, YES };
```

is better than

```
enum isWord{ YES, NO };
```

05. Tasks

Task-1

Create a structure called `time`. Its three members, all type `int`, should be called `hours`, `minutes`, and `seconds`. Write a program that prompts the user to enter a time value in hours, minutes, and seconds. This can be in 12:59:59 format, or each number can be entered at a separate prompt ("Enter hours:", and so forth). The program should then store the time in a variable of type `struct time`, and finally print out the total number of seconds represented by this time value:

```
long totalsecs = t1.hours*3600 + t1.minutes*60 + t1.seconds
```

Thank You!