Review Question:

In C an array is a complex data structure that can contain variables of different data types.

i. true

ii. false

Review Question:

In C a structure (struct) is a complex data structure that can contain variables of different data types.

i. true

ii. false

Review Question:

```
struct  PERSON
{
    int          nAge;
    double       fWeight;
    char cGender;
    char szName[10];
} Yang;
//assigning values to Yang...
```

How do you access the member variable cGender?

i. Yang -> cGender
ii. Yang.cGender
iii. *Yang
iv. Yang*cGender
v. cGender

Review Question:

```
struct PERSON
{
        int    nAge;
        double fWeight;
        char cGender;
        char szName[10];
};
```

```
struct STUDENT
{
        struct  PERSON Person;
        int nID;
        float fScore;
} Lee;
```

//assigning values to Lee...

How do I access Lee's nAge?

i. Lee.nAge
ii. nAge
iii. Lee.Person.nAge
iv. You can't have a structure inside a structure

Review Question:

Given PERSON   *pPerson;

How do I access the nID member?

i. pStudent->nID;
ii. nID
iii. pStudent*nID
iv. pStudent&nID

Typical Reasons for Using Pointers (1)

- When passing arguments to functions it may be preferable to pass a pointer, esp. to large amounts of data, rather than the actual data itself.

Typical Reasons for Using Pointers (2)

The implementation of some complex data structures (e.g. linked lists) requires the use of pointers.

Typical Reasons for Using Pointers (3)

When reading data from a file:

i. Identify how much data we hold in the file
ii. Allocate the correct amount of memory
iii. Copy the data into that memory
iv. Process the data
v. Write that data back to file

You need to be able to reference the relevant section of memory.

You can vary the size of the buffer, copying and deleting as you go.

Typical Reasons for Using Pointers (4)

- It is sometimes useful to return a pointer to relevant data, rather than the data itself, see

```
struct STUDENT* FindMinScoreStudent (int
            nNum, STUDENT *pStudent)
```

New Review Questions (1)

1. Declare a struct EMPLOYEE with the following members:

    struct PERSON Person,

    float Salary

    char job[10]

2. Print all members of PERSON Wang and Zhang.

3. Print all members of STUDENT Jin and Tian.

4. Why are we using strcpy to assign the name of a person:

    strcpy(Hou.Person.szName, "Hou");?

New Review Questions (2)

5. What does myClass[0] return?

6. What does myClass[1].fScore return?

7. Modify function PrintAllStudents(STUDENT *std, int numberOfStudents) so that it also prints the score and gender.

8. When is this condition false (evaluate to 0):

```
        if(fMinScore>pStudent->fScore)
```

9. Complete the function PrintAStudentPassingPointer(STUDENT *pStudent) so that it prints all details of a student.

New Review Questions (3)

10. Complete the function PrintAStudentPassingStruct(STUDENT student)

11. What does the function Birthday(STUDENT *pStd) do?

12. Is that change visible outside the function, after the function has finished running?

13. Why?

14. What is the type of pStudent in: (*pStudent).fScore?

15. What does this statement do: pStudent++; ?