

As minimal preparation for Quiz E, review the content related to Chapters 6 and 9 in:

- your textbook
- the COP 2000 Glossary pages for [Chapter 6](#) and [Chapter 9](#) (on the class web site)
- the Example pages for [Chapter 6](#) and [Chapter 9](#) (on the class web site)

---

**You will be asked questions related to C++ syntax or logic errors. The statements below provide examples of some typical errors. (Assume that all variables have been declared in advance.)**

SOURCE: float Ap; // Define a pointer to a float

ANSWER: The data type should be float \*

SOURCE: MyFunc (\*A); // Call MyFunc the address of scalar A

ANSWER: The "address of" operator is &, not \*

SOURCE: void MyFunc (\*); // Prototype receives passed int pointer \*/

ANSWER: Pointer data type names must include the data type, as in int \*

SOURCE: Ap=2; // Assign 2 into wherever pointer Ap points

ANSWER: Indirect referencing requires a unary \* operator in prefix notation (ahead of Ap).

SOURCE: MyFunc (A); // Call, passing the address of array A

ANSWER: NO ERROR: the use of an array identifier without subscript is equivalent to: &A[0]

**Know how to write and recognize valid C++ source code to perform each of the following tasks:**

- Define a pointer variable to hold the address of a double variable. For example:  
`double *Dptr;`
- Assign the address of a double variable named D into the double pointer named Dptr. Example:  
`Dptr = &D;`
- Pass the address of double variable D into a void function named Doit. For example:  
`Doit (&D);`
- Use a pointer to a double named Dptr to assign zero into the variable it points to. For example:  
`*Dptr = 0;`
- Increment the value pointed to by double pointer Dptr by 2. For example:  
`*Dptr = *Dptr + 2;`
- Increment the value of double pointer Dptr to point at the next double address in RAM. Example:  
`Dptr++;`

**You will be asked questions related to the passage of data by reference using pointers. For example, the following source code would pass a value into a function that will square the number and cube it and then return to values back to the main function to be displayed. Study the logic of the steps and the syntax of the statements.**

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

/* The function below receives one input parameter (N) and two output
   parameters (float pointers N2p and N3p to passed addresses) */
void SquareAndCube (float N, float *N2p, float *N3p)
{
    *N2p = N * N;      // Use indirection to assign the result to NUM2
    *N3p = *N2p * N;  // Use indirection to assign the result to NUM3
}

int main ()
{
    float NUM; /* Number to be processed */
    float NUM2; /* Number squared value */
    float NUM3; /* Number cubed value */

    cout << "Number to be processed? ";
    cin >> NUM;

    /* The function call below passes one input parameter (NUM) and two output
       parameters, the addresses (pointers to) NUM2 and NUM3, which allow them
       to be altered by the function using indirect assignment */
    SquareAndCube (NUM, &NUM2, &NUM3);

    cout << "Number: " << NUM << endl;
    cout << "Number squared: " << NUM2 << endl;
    cout << "Number cubed: " << NUM3 << endl;
    system ("pause");
    return 0;
}
```

The output produced by the code above would be (for the input value 2.5):

Number to be processed? 2.5 Number: 2.5 Number squared: 6.25 Number cubed: 15.625
--

(End of guide)