1. Mark the following statements as true or false.  
   a. The statement delete p; deallocates the variable pointer p.

false  
b. The statement delete p; deallocates the dynamic variable that is pointed to by p.

true  
c. Given the declaration:  
 int list [10];  
 int \*p;  
 the statement: p = list;  
 is valid in C++.

true  
d. Given the declaration:  
int \*p;  
the statement:  
p = new int[ 50 ] ;  
dynamically allocates an array of 50 components of type int, and p contains the base address of the array.

true

e. The address of operator returns the address and value of its operand.  
false

f. If p is a pointer variable, then the statement p = p \* 2; is valid in C++.

false

1. What does the following program do?  
   int i = 42;  
   int \*p1 = &i;  
   \*p1 = \*p1 \* \*p1;

Creates static memory i;

Defines pointer p1 and sets it the address of i;

Set pointer p1 to the value of dereferenced p1 multiplied by dereferenced p1

1. Explain each of the following definitions. Indicate whether any are illegal and, if so, why.  
   int i = 0;  
   a.double\* dp = &i;

Illegal, since a pointer has to be the same data type as what its pointing to

b. int \*ip = i;

illegal since a pointer cannot point to a value

c. int \*p = &i;

Setting pointer P to be the address of I, not illegal

1. What is the output of the following C++ code?  
   int x;  
   int y;  
   int \*p = &x;  
   int \*q = &y;  
   \*p = 35;  
   \*q = 98;  
   \*p = \*q;  
   cout << x << " " << y << endl;  
   cout << \*p << " " << \*q << endl;

98 98

98 98

1. Assume ip is a pointer to an int. Then, write a statement that will dynamically allocate an integer variable and store its address in ip. Write a statement that will free the memory allocated in the previous statement.

ip = new int;

delete ip;

1. Look at the following code.

int set[10];

Write a statement using pointer notation that stores the value 99 in set[7].

int set[10];

int \*ptr;

ptr = set;

ptr = ptr + 7;

\*ptr = 99;

1. Write code that dynamically allocates an array of 20 integers, then uses a loop to allow the user to enter values for each element in the array.

int \*ptr;

ptr = new int[10];

int number;

for(int i = 0; i<10;i++){

cin >> number;

\*(ptr + i) = number;

}

The following definitions and program segments have errors. Locate as many as you can.

1. int x, \*ptr;

&x = ptr;

1. int values[20], \*iptr;

iptr = values;

iptr \*= 2;

1. [Assume](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) that an int variable counter has already been declared. [Assume](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) further a variable counterPointer of [type](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) "pointer to [int](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)" has also [already](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) been [declared](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?). Write a [statement](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) that makes counterPointer "point" to counter.

counterPointer = &counter;

1. [Assume](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) that ip1, ip2, and ip3 have already been [declared](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) to be of [type](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) "pointer to int". [Assume](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) further that each of these pointer [variables](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) have been [initialized](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)-- each points to some [int](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) [variable](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?). Write a [statement](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) that computes the [sum](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) of the [variables](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) that ip1 and ip2 point to, and [assigns](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  that [value](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) (the [sum](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)) to the [variable](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?)  that ip3 points to.

\*ip3 = \*ip1 + \*ip2

1. What is the output of the following C++ code?  
   int \*length;  
   int \*width;  
   length = new int;  
   \*length = 5;  
   width = length;  
   length = new int;  
   \*length = 2 \* (\*width);  
   cout << \*length << " " << \*width << " " << (\*length) \* (\*width) << endl;

10 5 50

1. What is stored in list after the following code executes?  
   int list[ 7] = {10, 8, 15, 14, 16, 24, 36};  
   int \*ptr = list;  
   \*ptr = \*ptr + 2;  
   ptr = ptr + 2;  
   \*ptr = (\*ptr) – \*(ptr – 1);  
   ptr++;  
   \*ptr = 2 \* (\*ptr) – 3;

12,8,7,25,16,24,36

1. What is the output of the following C++ code?

int \*tempList;  
int num = 3;  
tempList = new int[7] ;  
tempList[6] = 4;  
for (int j = 5; j >= 0; j--)

tempList[j] = tempList[j + 1] + j \* num;  
for (int j = 0; j < 7; j++)

cout << tempList [j] << " ";  
cout << endl;

49 49 46 40 31 19 4

1. What is the output of the following C++ code?  
   int num;  
   int \*listPtr;  
   int \*temp;  
   listPtr = new int[5] ;  
   num = 8;  
   temp = listPtr;  
   for (int j = 0; j < 5; j++)  
   {

\*listPtr = num;

num = num + 2;

listPtr++;  
}  
listPtr = temp;  
for (int k = 0; k < 5; k++)  
{

\*temp = \*temp + 3;

temp++;  
}  
for (int k = 0; k < 5; k++)  
{

cout << \*listPtr << " ";

listPtr ++;  
}  
cout << endl;

11 13 15 17 19

1. Consider the following statement:  
   int \*num;  
   a. Write the C++ statement that dynamically creates an array of 10 components of type int and num contains the base address of the array.

Num = new int[10];

b. Write C++ code that inputs data into the array num from the standard input device.

std::cin >> \*num;

c. Write a C++ statement that deallocates the memory space of array to which num points.

delete [] num;

1. [Assume](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) that ip, jp, and tp have all been [declared](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) to be pointers to [int](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) and that result has been [declared](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) to be an [array](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) of 100 [elements](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?). [Assume](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) further that ip has been [initialized](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) to point to an [element](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) in the first half of the [array](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) and that jp has been [initialized](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) to point to an [element](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) in the second half of the [array](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?).

Write some code that makes jp point to the [element](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) that ip was pointing to and that makes ip point to the [element](http://pearson.turingscraft.com/codelab/jsp/core_dhtml.jsp?) that jp was pointing to.

Int temp = \*jp;

Jp = ip;

ip = &temp;

1. Given the declaration:  
   int num = 6;  
   int \*p = &num;  
   which of the following statements increment(s) the value of num?  
   a. p++;  
   b. (\*p)++;  
   c. num++;  
   d. (\*num)++;
2. What is the output of the following C++ code?  
   int \*p;  
   int \*q;  
   p = new int;  
   q = new int;  
   \*p = 27;  
   \*q = 35;  
   cout << \*p << " " << \*q << endl;  
   q = p;  
   \*q = 73;  
   cout << \*p << " " << \*q << endl;  
   p = new int;  
   \*p = 36;  
   \*q = 42;  
   cout << \*p << " " << \*q << endl;

27 35

73 73

36 42