Homework 9a

Gregory Louis

1. From the following code, what will be displayed if you send \*ptr to cout? What if you send ptr to cout?

int x = 7;

int \*ptr = &x;

The value 7 will be displayed if the expression \*iptr is sent to cout.  
If the expression iptr is sent to cout, the address of the variable x will be displayed.

.

1. What will the following display?

int numbers[] = {2, 4, 6, 8, 10};

cout << \*(numbers + 3) << endl;

8

.

1. Give at least two uses of the \* operator. State what the \* is doing, and name the use of the \* that you present.

Int \*p; this declares a pointer variable that can hold a pointer to an int variable. Hold a pointer to an int variable.

\*p = 17; here, \* is the dereference operator , this assigns 17 to the memory location pointed to by p.

1. What is the output produced by the following code?

*int* \*p1, \*p2;

p1 = *new int*;

p2 = *new int*;

\*p1 = 10;

\*p2 = 20;

cout << \*p1 << " " << \*p2 << endl;

p1 = p2;

cout << \*p1 << " " << \*p2 << endl;

\*p1 = 30;

cout << \*p1 << " " << \*p2 << endl;

How would the output change if we replace \*p1 = 30; with the following? \*p2 = 30;

10 20  
20 20  
30 30

The output would be the same if we replace \*p1 = 30; , with \*p2 = 30.

1. Suppose a dynamic variable were created as follows:

*char* \*p;

p = *new char*;

Assuming that the value of the pointer variable p has not changed (so it still points to the same dynamic variable), how can you destroy this new dynamic variable and return the memory it uses to the freestore so that the memory can be reused to create new dynamic variables?

char \*p;

p = new char

above code will allocate memory for a character and store the address of allocated storage to variable p

This technique of memory allocation is called dynamic memory allocation

and to deallocate dynamically allocated memory use delete keyword as below

delete pointer;

so for this case

**delete p;** will deallocate the storage to which p is pointing

.

1. What is the output produced by the following code?

*int* \*p1, \*p2;

p1 = *new int*;

p2 = *new int*;

\*p1 = 10;

\*p2 = 20;

cout << \*p1 << " " << \*p2 << endl;

\*p1 = \*p2; //*This is different from Exercise 4*

cout << \*p1 << " " << \*p2 << endl;

\*p1 = 30;

cout << \*p1 << " " << \*p2 << endl;

10 20  
20 20  
30 20

.

1. Write a definition for a type called NumberPtr that will be the type for pointer variables that hold pointers to dynamic variables of type *int*.   
   Also, write a declaration for a pointer variable called my\_point that is of type NumberPtr.

Typedef int\* NumberPtr;

NumberPtr my\_point;

1. Describe the action of the *new* operator. What does the operator *new* return?

The new operator takes a type for its argument. New allocates space on the freestore of an appropriate size for a variable of the type of the argument. It returns a pointer to that memory (that is, a pointer to that new dynamic variable), provided there is enough available memory in the freestore. If there is not enough memory available in the freesotre, your program ends.

1. Write a cout statement that uses the ptr variable to display the contents of the value variable in the following:

double value = 29.7;

double \*ptr = &value;

cout<< \*ptr;// uses the ptr variable to display the contents of the value variable.

.

1. In the following code change n from being a reference variable to a pointer:

void getNumber (int &n)

{

cout << “Enter a number: “;

cin >> n;

}

void getNumber(int \*pn) .