Answers to Review Questions
Chapter 9

1. It dereferences a pointer, allowing code to work with the value that the pointer points to.

2. 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.

3. Multiplication operator, definition of a pointer variable, and the indirection operator.


5. It adds 4 times the size of an int to the address stored in ptr.

6. 8

7. To dynamically allocate memory.

8. An exception is thrown, which causes the program to terminate. Under older compilers, the new operator returns the null address (address 0) when it cannot allocate the requested amount of memory.

9. To free memory that has been dynamically allocated with the new operator.

10. You should only return a pointer from a function if it is
    - A pointer to an object that was passed into the function as an argument
    - A pointer to a dynamically allocated object

11. address
12. address (&)
13. pointer
14. indirection (*)
15. pointers
16. dynamic memory allocation
17. new
18. 0 or null
19. null
20. delete
21. new

22. cout << *ptr << endl;

23. *(set + 7) = 99;

24. int *ptr;
ptr = new int[20];
for (int i = 0; i < 20; i++)
{
    cout << "Enter a value: ";
    cin >> ptr[i];
}

delete [] tempNumbers;

void getNumber(int *n)
{
    cout << "Enter a number: ";
    cin >> *n;
}

true
false
true
false
false
true
false
false
true
true
false
true
false
false
true
false
The variable should be declared as int *ptr;
The assignment statement should read ptr = &x;
The assignment statement should read ptr = &x;
The assignment statement should read *ptr = 100;
The last line should read
    cout << *(numbers + 3) << endl;
Multiplication cannot be performed on pointers.
An int pointer is being used to reference a float value.
iptr cannot be initialized with the address of ivalue. ivalue is defined after iptr.
The * operator is used on the parameter, but it is not a pointer.
The second statement should read pint = new int;
The program segment is storing a value at the address 0.
The last line should read delete [] pint;
56. The function returns the address of a variable that no longer exists.