1. Declare a pointer variable named ptr to an integer.

2. Write the code that assigns to p1 (an integer pointer variable) the pointer to a dynamically created integer.

3. Write the code to return the dynamic memory pointed to by p1 to the freestore.

4. Write the code to declare a dynamic array of strings (use the string pointer variable p1) that has as many elements as the variable arraySize.

5. Which of the following correctly declare 3 integer pointers?
   a. int* p1, p2, p3;
   b. int *p1, p2, p3;
   c. int *p1, *p2, *p3;
   d. all of the above.

6. Which of the following assigns to p1 the pointer to the address of value?
   a. *p1=&value;
   b. p1=value;
   c. p1=&value;
   d. &p1 = *value;

7. What is the output of the following code fragment?
   int v1=2, v2=-1, *p1, *p2;
   p1 = &v1;
   p2= & v2;
   p2=p1;
   cout << *p2 << endl;
   a. 2
   b. -1
   c. -2
   d. 1

8. Which of the following statements correctly prints out the value that is in the memory address that the pointer p1 is pointing to?
   a. cout << &p1;
   b. cout << p1;
   c. cout << int* p1;
   d. cout << *p1;

9. Given that p1 is a pointer variable of the string class, which of the following are legal statements?
   a. p1 = new int;
   b. cout << *p1;
   c. p1 = new char[10];
   d. *p1 = new string;
10. What is the output of the following code fragment?
float *p1;
p1 = new float(3);
cout << *p1;
   a. 3.0
   b. unknown, the address p1 points to is not initialized
   c. unknown, the code is illegal, p1 points to a dynamic array
   d. 0.0

11. What is the output of the following code?
int *p1, *p2;
p1 = new int;
p2 = new int;
*p1=11;
*p2=0;
p2=p1;
cout << *p1 << " " << *p2 << endl;
   a. 11 0
   b. 0 11
   c. 11 11
   d. 0 0

12. Which of the following correctly declares a user-defined data type that defines a pointer to a float?
   a. float* floatPtr ;
   b. typedef float* floatPtr;
   c. typedef floatPtr *float;
   d. typedef floatPtr* float

13. Which of the following correctly declares a dynamic array of strings?
   a. p1 = new string(13);
   b. p1 = new string[];
   c. p1 = new string[13];
   d. p1 = new stringArray(13);

14. What is wrong with the following code fragment?
int *p1, *p2;
p1 = new int;
p2 = new int;
*p1=11;
*p2=0;
p2=p1;
cout << *p1 << " " << *p2 << endl;
delete p1;
delete p2;
   a. nothing
b. p1 and p2 both have the same value, so the delete p2 will cause an error

c. You have a memory leak.

d. B and C

15. Given that a typedef for IntPtr defines a pointer to an integer, what would be the correct declaration for a function that expects a reference to an integer pointer?
   a. void f1 (IntPtr& ptr);
   b. void f1 (IntPtr&* ptr);
   c. void f1 (IntPtr* & ptr);
   d. All of the above

16. Given that p1 is an integer pointer variable, and a1 is an integer array, which of the following statements are not legal code?
   a. p1 = a1;
   b. cout << p1[0];
   c. cin >> p1[0];
   d. a1 = p1;

17. Assuming that the pointer variable p1 is of the correct type and size is an integer with some value > 1, which of the following declarations are legal?
   a. p1 = new string[size];
   b. p1 = new ifstream[size];
   c. p1 = new char[size*size];
   d. A and B
   e. A, B and C

18. Which of the following statements correctly returns the memory from the dynamic array pointer to by p1 to the freestore?
   a. delete [] p1;
   b. delete p1[];
   c. delete *p1;
   d. delete p1;

19. If p1 is an integer pointer that is pointing to memory location 1001, and an integer takes 4 bytes, then (p1+1) evaluates to:
   a. 1002
   b. 1004
   c. 1005
   d. Unknown

20. Given that p1 is a pointer, p1++
   a. always causes a run time error
   b. advances p1 by one unit of the type of variable to which p1 points
   c. adds 1 to whatever p1 is pointing to
   d. adds one element to the array that p1 is pointing to

21. If a program requires a dynamically allocate two-dimensional array, you would allocate the memory by using
   a. p1 = new int*[numRows];
for(int i=0; i < numRows; i++)
    p1[i] = new int[numColumns];

b. p1 = new int*[numRows][numColumns];
c. p1 = new[numRows][numColumns]int;
d. none of the above

22. If two pointer variables point to the same memory location, what happens when one of the pointers is freed?
   a. The other pointer should be considered to be un-initialized
   b. The other pointer still points to a valid memory address
   c. If you attempt to free the other pointer a run-time error will occur.
   d. All of the above
   e. A and C

23. Which of the following is not true?
   a. a pointer can be assigned the address of an array
   b. an array can be assigned the value in a pointer variable
   c. if a pointer points to an array, it can be used in place of the array name
   d. if a pointer points to an array, the pointer does not know how many elements are in the array.

24. In which case would you consider using a dynamic array?
   e. If the array is small, and the size is known before the program runs.
   f. If the program needs to get the size of the array from the user
   g. If the array size is big, but known at compile time
   h. You should always use a dynamic array