1. Why is a static local variable useful?
   A. It is created outside of any function.
   B. It retains its value between function calls.
   C. It is initialized on each function call.
   D. Its value cannot be accidentally changed.

2. If a function doesn't return a value, what is its return type?
   A. none
   B. null
   C. void
   D. bool

3. Given the following function header, how can your program call the function?
   ```
   void showValue(int quantity)
   ```
   A. `cout << showValue(10);`
   B. `showValue(int my_quantity);`
   C. `call showValue(100);`
   D. `showValue(sum);`
   E. `quantity = showValue();`

4. A class may have more than one constructor, as long as each has a
   A. destructor
   B. different parameter list
   C. different name
   D. unique return type

5. If you see this line of code in a C++ program. What do you know?
   (The program has no structs defined)
   ```
   Vehicle car(2016);
   ```
   A. There is a constructor named car that accepts an argument.
   B. There is a class instance named car.
   C. There is a class definition named car.
   D. There is a class named Vehicle with at least one constructor defined.
   E. There is a class named Vehicle with a member function named car.
   F. There is a member variable that stores the car's year.

6. When is a copy constructor called?
   A. When a class is passed as a by-value argument to a function.
   B. When a class is passed as a by-reference argument to a function.
   C. Whenever the assignment operator `=` is used.
   D. When the class has no pointer variables defined.
7. Given the following array definition:

```c
int values[5] = { 4, 7, 6, 8, 2 };
```

What does each of the following statements display?

- `cout << values[4];` ________ 2
- `cout << (values[2] + values[3]);` ________ 14
- `cout << ++values[1];` ________ 8

8. Show the output of the following code.

```c
int numbers[] = { 2, 4, 6, 8, 10 };
cout << *(numbers + 3) << endl;
```

8

9. What is the exact output of the following code segment? Circle your answer.

```c
int a=5, b=4, c;
c = ++a;
b = c++;
a = a + b - c;
cout << a << b << c;
```

567

10. Look at the following code, then write one statement that will display the value of variable x without using the variable x.

```c
int x = 7;
int *ptr = &x;
cout << *ptr;
```

11. The array below is used to store an image of resolution 640x480 pixels. The image is 640 pixels across and 480 pixels high. The following one-dimensional array stores each pixel of the image. Write one or more statements that will set the pixel in the 3rd row and 4th column to a value of 255.

```c
int arr[480*640];
arr[2*640+3] = 255;
```

```c
skip first 2 rows = 640*2 = 1280
skip 3 more = 3
----
total 1283
```

No students got this question correct. Some partial credit given.
12. Look at the class definition below. Write one statement that will declare your own instance of the class. Then use cout or printf, write one statement that will display the data-member values of the class you declared.

```cpp
class Shape {
    public:
        int height;
        int width;
} box;
```

one solution is below:
```cpp
Shape rect;
cout << rect.height << " " << rect.width;
```

13. Using polymorphism requires what C++ elements? (1-pt each)

   A. a class
   B. an array
   C. a pointer variable <--- pointer to the derived class is required
   D. a virtual function
   E. a cout statement
   F. inheritance
   G. a static variable
   H. memberwise assignment
   I. a derived class
   J. all private access

14. Memberwise assignment

   A. completely copies one class to another <--- not completely always
   B. does not copy a pointer's address value. <--- address only is copied
   C. is the function of a default copy constructor
   D. requires overloading the copy constructor.

15. Other than C++ or a Java language, name two languages that are considered to be object-oriented.

   1. C#  2. Smalltalk

   and others we saw in class

16. T  F  Class members are private by default.

17. T  F  It is legal to define a pointer to a class object.

18. T  F  An instance of a class can be dynamically allocated in memory.

19. T  F  Overloaded constructors must have a return type defined.

20. T  F  Destructors can be overloaded.

21. T  F  The [] operator cannot be overloaded in a class.

   The [] operator can be overloaded, as it is in the STL String class.
22. Write code that is neat, clean, and correct please. (10 pts)

a. Declare an array that can be used to store exactly 125 integer values.
   ```cpp
   int arr[125];
   ```
   (not double!)

b. Using one line of code, declare and initialize a pointer variable that points to the start of your array.
   ```cpp
   int *p = arr;
   ```

c. Using just two lines of code, write a for-loop that will fill your array with unique odd numbers greater than or equal to 5.
   ```cpp
   for (int i=0; i<125; i++)
       arr[i] = i*2+5;
   ```
   several other ways to do this. arr[i] = i+i+5;

d. Write a complete C++ class that will do the tasks above given the main function below.
   ```cpp
   int main() {
       MyList list(125);
       list.fill_with_odd(5);
       return 0;
   }
   ```
   Write your class here:

   one solution is below:

   ```cpp
   class MyList {
       int *arr;
       int size;
   public:
       MyList() {
           size = 0;
           arr = NULL;
       }
       MyList(int n) {
           size = n;
           arr = new int[size];
       }
       void fill_with_odd(int num) {
           num = num | 1; //set LSB to 1. Other ways to do this.
           for (int i=0; i<size; i++)
               arr[i] = (i*2) + num;
       }
       ~MyList() { delete [] arr; }
   };
   ```

   The array should be dynamically allocated, as indicated by the constructor's argument in main. Size should be saved for use in the for-loop. Function can adjust for an even number passed in, but not required.