1. In the following program segment
   ```c
   #ifndef X
   rest of program
   #endif
   ```
   (a) will evaluate the rest of the program if X is already defined.
   (b) will evaluate the rest of the program if X is not already defined.
   (c) will evaluate the rest of the program regardless of whether X is defined.
   (d) will cause a syntax error.
   ANS: (b)

2. Constructors are not
   (a) required to be explicitly defined.
   (b) called automatically when an object is initialized.
   (c) able to be overloaded.
   (d) member functions.
   ANS: (a)

3. A class may contain multiple constructors if
   (a) they have different names.
   (b) they have different argument lists.
   (c) they have the same argument list.
   (d) they have different return types.
   ANS: (b)

4. A default constructor
   (a) is a constructor with all default arguments
   (b) is the constructor generated by the compiler when one is not provided by the programmer
   (c) does not perform any initialization
   (d) both (b) and (c)
   ANS: (d)

5. Which of the following is not true of a constructor and destructor of the same class?
   (a) they both have same name aside from the tilde (~) character.
   (b) they are both called once per object (in general).
   (c) they both are able to accept default arguments.
   (d) both are called automatically, even if not defined in the class.
   ANS: (c)

6. Assume class CreateAndDestroy has a constructor and a destructor function that acts as follows:
   When an object of this class is created its constructor takes two strings as arguments and prints a message. For example, the statement `CreateAndDestroy c("c", "In Example")` would call its constructor, which will simply print out the message “Created c: In Example”. Its destructor will, on the other hand simply print out the message “Destroyed c: In Example”. Now consider the following code sequence.
Write the sequence of creation and destruction of objects if the program containing above code is run.

```c
main()
{
    CreateAndDestroy cc1("c1", "auto - main");
    static CreateAndDestroy sc1("sc1", "static - main");
    firstThingFirst("f-first");
    static CreateAndDestroy sc2("sc2", "static - main");
}

firstThingFirst(chr *msg)
{
    static i = 1;
    CreateAndDestroy fc1(msg, "auto - firstThingFirst");
    static CreateAndDestroy fscl(msg, "static - firstThingFirst");
    for (; i < 3; i++){
        switch (i) {
            case 1:
                static CreateAndDestroy fsc("case 1", "static -
                firstThingFirst");
                firstThingFirst("f-case1");
                break
            case 2:
                static CreateAndDestroy fsc("case 2", "static -
                firstThingFirst");
                firstThingFirst("f-case2");
                break;
        default: break;
    }
}
```

Answer

```
MAIN FUNCTION: EXECUTION BEGINS
Object 1 constructor runs <local automatic in main>
Object 2 constructor runs <local static in main>

CREATE FUNCTION: EXECUTION BEGINS: 2
Object 10 constructor runs <Here: local automatic in create>
Object 14 constructor runs <static - switch case 2>

CREATE FUNCTION: EXECUTION BEGINS: 1
Object 4 constructor runs <Here: local automatic in create>
Object 6 constructor runs <static - switch case 1>

CREATE FUNCTION: EXECUTION BEGINS: 0
Object 0 constructor runs <Here: local automatic in create>

CREATE FUNCTION: EXECUTION ENDS: 0
Object 0 destructor runs <Here: local automatic in create>

CREATE FUNCTION: EXECUTION ENDS: 1
Object 4 destructor runs <Here: local automatic in create>

CREATE FUNCTION: EXECUTION ENDS: 2
Object 8 destructor runs <Here: local automatic in create>

MAIN FUNCTION: EXECUTION ENDS
Object 1 destructor runs <local automatic in main>
Object 6 destructor runs <static - switch case 1>
Object 14 destructor runs <static - switch case 2>
Object 10 destructor runs <static in create>
Object 2 destructor runs <local static in main>
Press any key to continue...
```