{{{questionNumber}}}. Consider this simple example.

```
int * a;
int * b; a
= new
int(5); b =
a;
cout << *b <<
endl; delete
a; a = NULL; b
= NULL;
```

What is the result of executing these statements if you assume the standard `iostream` library has been included?

A. None of the other options describes the behavior of this code.
B. This code does not compile.
C. [Correct Answer] [Your Answer] `5` is sent to standard out and no memory is leaked.
D. The memory address of `b` is sent to standard out.
E. This code has a memory leak.
F. This code results in undefined runtime behavior.

{{{questionNumber}}}. Consider this simple example.

```
int * p; int
i; i = 37; *p
= i; *p = 99;
cout << i <<
endl;
```

What is the result of executing these statements, assuming that `iostream` is included?

A. This code has a memory leak.,
B. None of the other options describes the behavior of this code.
C. This code does not compile.
D. [Correct Answer] This code results in undefined runtime behavior.
E. `37` is sent to standard out.
F. [Your Answer] `99` is sent to standard out.

```
#include
<iostream> using
namespace std;

class Bear
{     public:
        Bear() { cout << "Growl "; }
        ~Bear() { cout << "Stomp stomp stomp "; }
};

int main()
{     Bear
beary;     cout
<< "Run! ";
return 0;
}
```

{{{questionNumber}}}. What is the result of compiling and executing this code?

A. [Correct Answer] `Growl Run! Stomp stomp stomp`
B. This code does not compile.
C. `Run! Stomp stomp stomp`
D. [Your Answer] `Growl Run!`
E. `Run!`

{{{questionNumber}}}. Consider this simple example

```
class Pumpkin
{     public:
        Pumpkin(double radius, int * seeds)
        Pumpkin(const Pumpkin & other);
        ~Pumpkin();
        // more public member functions

    private:
double radius;
int *seeds;
        // more private member
variables };
```

Which of the following functions must also be implemented for the `Pumpkin` class for it to function correctly?

A. [Your Answer] No Parameter Constructor
B. `setRadius()`
C. `operator delete`
D. [Correct Answer] `operator=`

E. `operator()`

{{{questionNumber}}}. Which of the following is a correct way to initialize the variable named `NCC1701` to be a dynamic array of `starShip` pointers with size `size`?

A. `starShip * [size] NCC1701;`
B. None of the other answers are correct initializations for `NCC1701`.
C. `starShip * NCC1701[size];`
D. `NCC1701 = new starShip[size];`
E. `for (int i = 0; i < size; i++) NCC1701[i] = new starShip *;`
F. [Correct Answer] [Your Answer] `NCC1701 = new starShip *[size];`