NetID: roatis2 QuizID: 98294 Score: 2 / 5 Answer Source: Manual Input from Quiz Sheet

```
class puppy {
    public:
        puppy() { cout << "yip "; }
        puppy(puppy const & orig) { cout << "bounce "; }
        ~puppy() { cout << "arf "; }
};

void walk(puppy const & p) {
    cout << "prance ";
}

int main() {
    puppy pete;
    walk(pete);
    return 0;
}
```

{{{questionNumber}}}. What is the result of compiling and executing this code, assuming that `iostream` is included?

A. `yip`
B. None of the other answers is the output.
C. [Correct Answer] `yip prance arf`
D. `yip bounce prance`
E. `prance`
F. [Your Answer] `yip bounce prance arf`

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

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

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

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

{{{questionNumber}}}. Which of the following is a correct function signature for the overloaded addition operator for the `sphere` class, if we want that operator to return a `sphere` whose radius is the sum of the radii of the object and its parameter?

A. `sphere sphere::operator+(const sphere & left, const sphere & right);`
B. `sphere & sphere::operator+();`
C. More than one of the three function signatures, could be used.
D. None of the other options is appropriate
E. [Correct Answer] [Your Answer] `sphere sphere::operator+(const sphere & right) const;`

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

A. `starShip * NCC1701 = new starShip(NCC1701);`
B. `starShip * [size] NCC1701;`
C. [Your Answer] None of the other answers are correct declarations for `NCC1701`.
D. [Correct Answer] `starShip ** NCC1701;`
E. `starShip * NCC1701 = new starShip *[size];`

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