

{{{questionNumber}}}. What does the following code print out?

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

int main() {
    int *a = new int[2];
    for(int i = 0; i < a.length(); i++) {
        cout << a[i];
    }
    return 0;
}
```

- A. The code will segfault.
- B. [Your Answer] Two garbage values.
- C. 00
- D. [Correct Answer] The code does not compile.
- E. None of the other answers is true.

{{{questionNumber}}}. Consider this simple code, and assume the puppy class has default and copy constructors defined:

```
puppy * plantANew(puppy orig) {
    puppy * seedling = new puppy(orig);
    return seedling;
}

int main() {
    puppy f1; puppy * f2;
    f2 = plantANew(f1);
    return 0;
}
```

How many times is a puppy constructor called in the example above?

- A. [Your Answer] One time.
- B. Twice.
- C. Never, but the code executes with no errors.
- D. [Correct Answer] Three times.
- E. Never, because this code has a compiler error.

{{{questionNumber}}}. Consider the following code:

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

void myfunc(int y, int *x) {
    y = y+1;
    cout << *x << endl;
    y = y+1;
    *x = y;
}

int main() {
    int z = 6;
    int *x = &z;
    myfunc(z, x);
    myfunc(z+1, x);
    return 1;
}
```

What is the result of compiling and running this code?

- A. Nothing is printed to the screen.
- B. This code has a compilation error.
- C. [Correct Answer] The numbers 6 and 8 are printed to the screen.
- D. [Your Answer] The numbers 7 and 10 are printed to the screen.
- E. The numbers 6 and 9 are printed to the screen.

{{{questionNumber}}}. Consider this simple function definition.

```
PNG & ugly(PNG x) {
    return x;
}
```

Which of the following statements is true?

- A. [Your Answer] This function is ugly because it returns a value parameter by reference.
- B. This function is ugly because the parameter is not PNG const x.
- C. This function is ugly because it could be slow.
- D. [Correct Answer] This function is ugly for two of the other reasons.
- E. This function is ugly because there is a type mismatch between the return value and the return type.