

Agenda / Learning Objectives:

1. Discuss the answer for question 2 in quiz 5
2. Discuss how to track the variable updates in lab 18 questions 4 and 5
3. Apply learning strategies covered in [TeachYourselfHowToLearn.pptx](#) and EC2
4. Answer question 1 in professor Ryba's recursion exercises

From Lab 18 – functions (passed by reference)

Analyze a program for output:

4) (prac2.pdf) Consider the following C++ program.

```
#include <iostream>
using namespace std;
int fun(int &x, int &y) {
    if (y <= 0) return x;
    x = x + 2;
    cout << x << y << endl;
    return x * y;
}
int main() {
    int x = 4, y = 0;
    cout << fun(x, y) << endl;    // line a
    fun(y, x);                    // line b
    fun(x, y);                    // line c
    fun(y, x);                    // line d
    cout << fun(x, y) << endl;    // line e
    return 0;
}
```

What is the output from the program at each of the following lines:

- (a) line a:
- (b) line b:
- (c) line c:
- (d) line d:
- (e) line e:

5) (prac2.pdf) Consider the following C++ program.

```
#include <iostream>
using namespace std;
int fun(int &x, int y) {
    x = x + 1;
    y = y - 1;
    return y;
}
int main() {
    int x = 2, y = 7, z = 10; string s = "007";
    cout << ((double) y) / x << endl;    // line (a)
    if (!(x > y) && (y > 5)) s = "008";
    cout << s << endl;                    // line (b)
    z %= y; cout << z << endl;            // line (c)
    cout << fun(z, y) << endl;            // line (d)
    fun(x, y); cout << y - x * 2 << endl; // line (e)
}
```

- (a) What is the output at line (a)?
- (b) What is the output at line (b)?
- (c) What is the output at line (c)?
- (d) What is the output at line (d)?
- (e) What is the output at line (e)?

(From lab 18) Title Lines:

7) (prac2.pdf & prac3.pdf) Write the best **title lines** for the functions that are called by the following main program. **Do not supply blocks for the functions.**

```
int main() {  
    int x = 0, y = 1, z = 2;  
    x = sum(z, y);           // (a) sets x to the sum: 3  
    reset(y, z);           // (b) replaces y by the value of z  
    makeNegative(z);       // (c) make z negative  
    boost(x, y);           // (d) increase x by the value of y  
    boost(y, mystery(y, z)); // (e) boosts y by a mystery amount  
    return 0;  
}
```

(a) Title line for sum.

(b) Title line for reset.

(c) Title line for makeNegative.

(d) Title line for boost.

(e) Title line for mystery.