

## Part 1: Warm-up & Program Analysis

Code:

```
#include <iostream>

int main() {
    int a = 5, b = 12, c = 2;
    double x = 10.0, y = 3.0;

    std::cout << "Result 1: " << b / a << std::endl;
    std::cout << "Result 2: " << b % a << std::endl;
    std::cout << "Result 3: " << x / y << std::endl;
    std::cout << "Result 4: " << (a + b + c) / y << std::endl;
    std::cout << "Result 5: " << static_cast<double>(b) / a << std::endl;

    return 0;
}
```

1. What will be the exact output for "Result 1"? Why isn't it 2.4?

A and B are both set as an integer, which doesn't show the decimal place. Therefore, the answer would be just 2.

2. What will be the exact output for "Result 2"? What does the % operator do?

The output for "Result 2" would be 2, and the % operator shows the remainder of the numbers that are divided. In this case it is  $12 \% 5$  which is the remainder of  $12/5$ , so it would be 2.

3. What will be the exact output for "Result 3"?

3.3333333 and the 3 will be repeating for ever, because x and y are both double which shows the decimal place as well.

4. What will be the exact output for "Result 4"?

6.333333, 3 being repeated like result 3.

5. What will be the exact output for "Result 5"? How does static\_cast change the calculation?

The answer would be 2.4 and static\_cast will make "b" into double, which was originally int. If you change the type to double it will show you the decimal place as well.