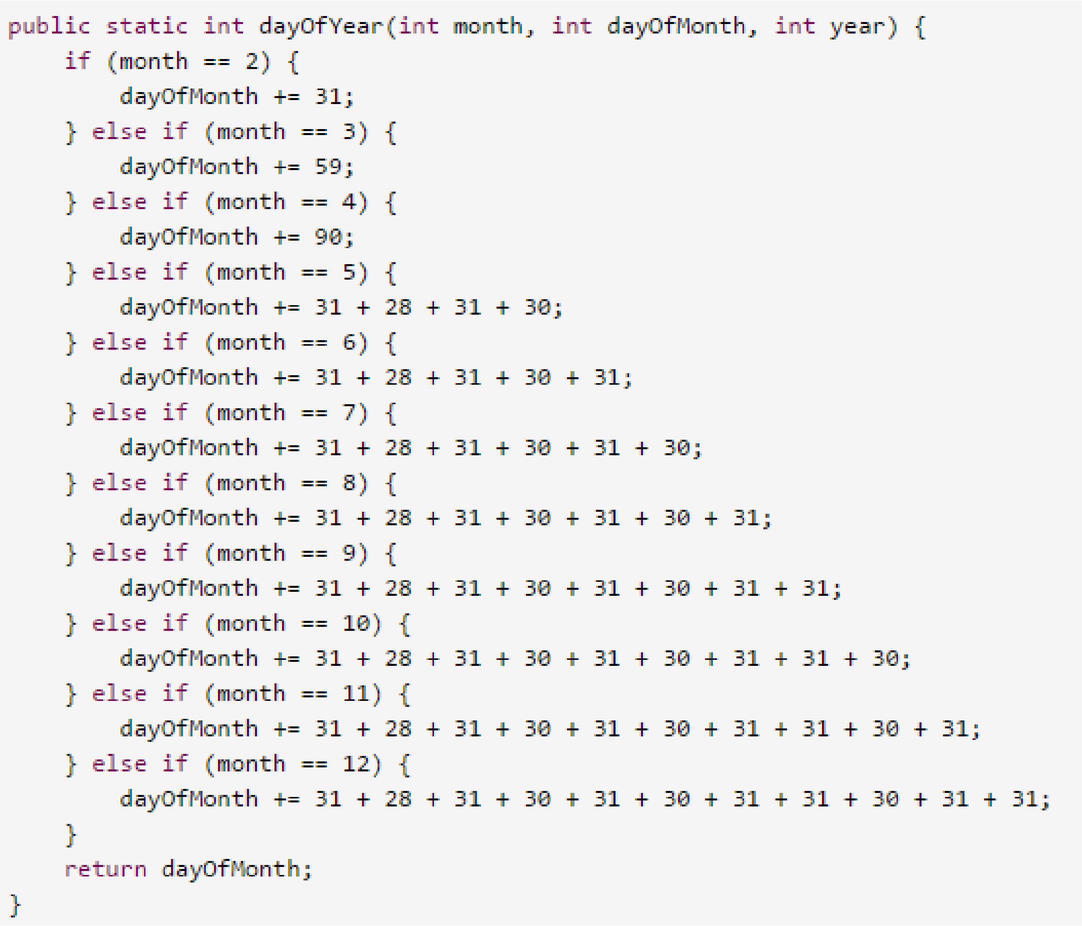
Software Testing and Quality Assurance

COMP438

Tutorial # 3: Code Inspection

Objective: Practice general principles of good coding, which you can look for in every code review, regardless of programming language or program purpose. Carefully inspect the following piece of code and try answering the following questions

PART 1



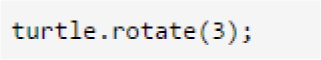
```
public static int dayOfYear(int month, int dayOfMonth, int year) {
    if (month == 2) {
        dayOfMonth += 31;
    } else if (month == 3) {
        dayOfMonth += 59;
    } else if (month == 4) {
        dayOfMonth += 90;
    } else if (month == 5) {
        dayOfMonth += 31 + 28 + 31 + 30;
    } else if (month == 6) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31;
    } else if (month == 7) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30;
    } else if (month == 8) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31;
    } else if (month == 9) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31;
    } else if (month == 10) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30;
    } else if (month == 11) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31;
    } else if (month == 12) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31 + 31;
    }
    return dayOfMonth;
}
```

Part 1 Discussion

- Inspect the above code carefully can reveal several types of errors from different angles/perspectives. Can you identify these perspectives in general?
- What is wrong with the above code? Would you approve it?
- Is there a better way to design it?

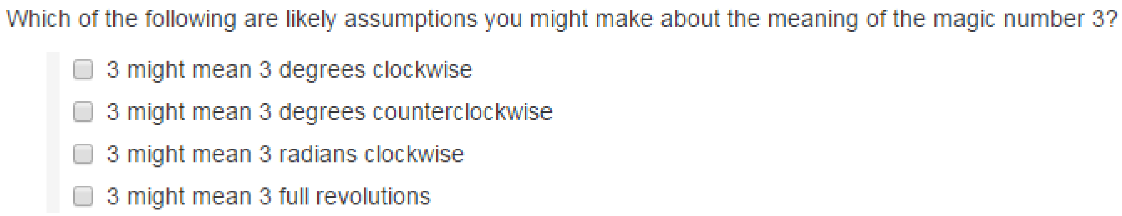
Part 2:

Suppose you're reading some code that uses a turtle graphics library that you don't know well, and you see the code:



```
turtle.rotate(3);
```

Discussion

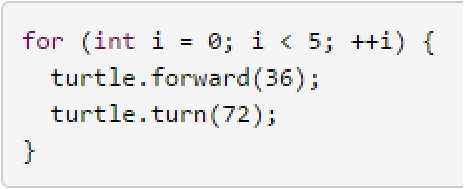


Which of the following are likely assumptions you might make about the meaning of the magic number 3?

- □ 3 might mean 3 degrees clockwise
- □ 3 might mean 3 degrees counterclockwise
- □ 3 might mean 3 radians clockwise
- □ 3 might mean 3 full revolutions

PART 3:

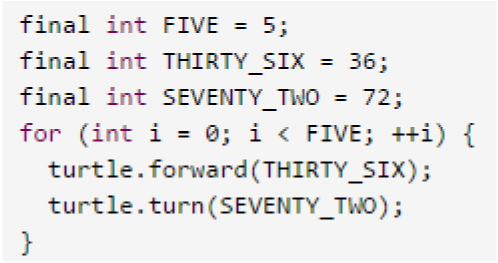
Consider this code:



```
for (int i = 0; i < 5; ++i) {
  turtle.forward(36);
  turtle.turn(72);
}
```

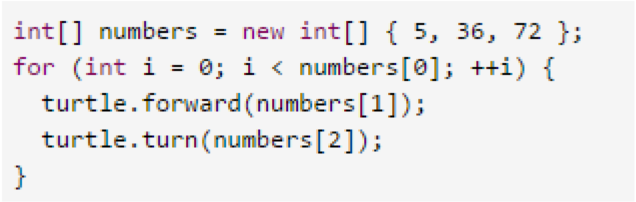
For each of the following re-writes, which one you would choose and **WHY**?

A)



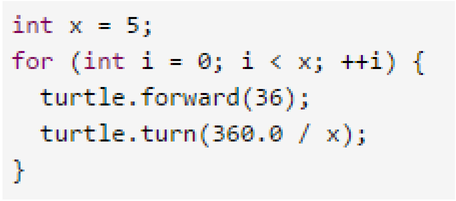
```
final int FIVE = 5;
final int THIRTY_SIX = 36;
final int SEVENTY_TWO = 72;
for (int i = 0; i < FIVE; ++i) {
  turtle.forward(THIRTY_SIX);
  turtle.turn(SEVENTY_TWO);
}
```

B)



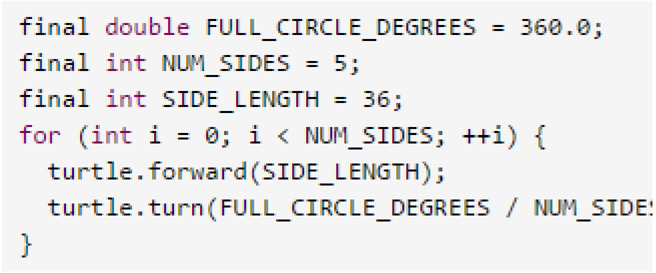
```
int[] numbers = new int[] { 5, 36, 72 };
for (int i = 0; i < numbers[0]; ++i) {
  turtle.forward(numbers[1]);
  turtle.turn(numbers[2]);
}
```

C)



```
int x = 5;
for (int i = 0; i < x; ++i) {
  turtle.forward(36);
  turtle.turn(360.0 / x);
}
```

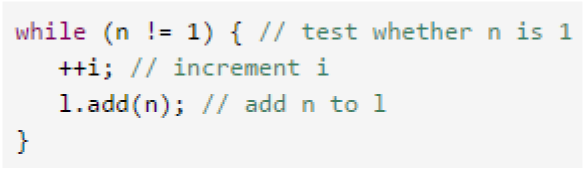
D)



```
final double FULL_CIRCLE_DEGREES = 360.0;
final int NUM_SIDES = 5;
final int SIDE_LENGTH = 36;
for (int i = 0; i < NUM_SIDES; ++i) {
  turtle.forward(SIDE_LENGTH);
  turtle.turn(FULL_CIRCLE_DEGREES / NUM_SIDE
}
```

PART 4:

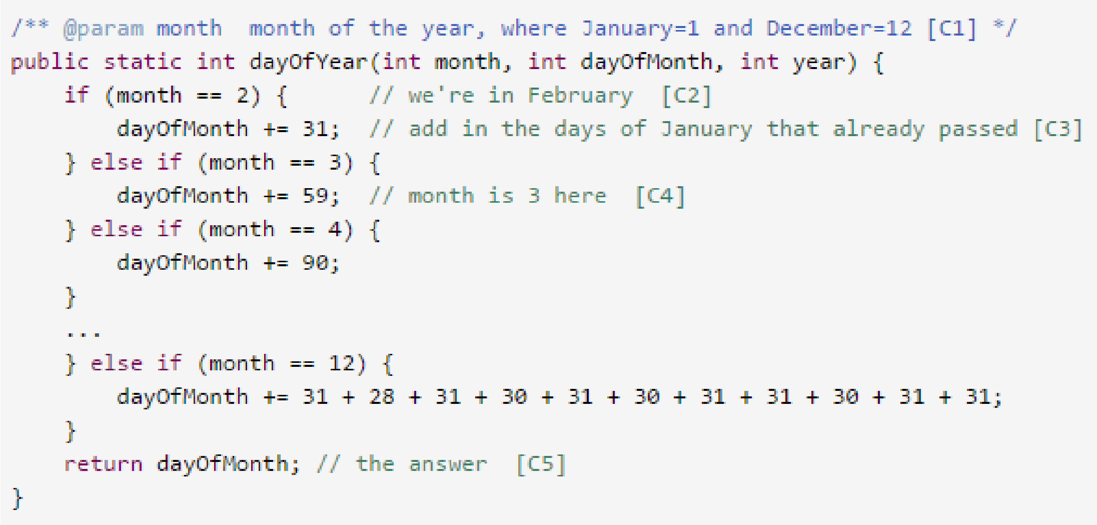
What do you think about the following comments? What should be really written when we write down comments for our code?



```
while (n != 1) { // test whether n is 1
   ++i; // increment i
   l.add(n); // add n to l
}
```

Part 5

Which comments are useful additions to the code? Consider each comment independently, as if the other comments weren't there.



```java
/** @param month  month of the year, where January=1 and December=12 [C1] */
public static int dayOfYear(int month, int dayOfMonth, int year) {
    if (month == 2) {      // we're in February  [C2]
        dayOfMonth += 31;  // add in the days of January that already passed [C3]
    } else if (month == 3) {
        dayOfMonth += 59;  // month is 3 here  [C4]
    } else if (month == 4) {
        dayOfMonth += 90;
    }
    ...
    } else if (month == 12) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31 + 31;
    }
    return dayOfMonth; // the answer  [C5]
}
```