Week 3 Reflection

Samuel Mitchell - Quality Control

Charles Ceccardi - Facilitator

Andrew Martin - Spokesperson

Mitchell Oneka - Process Analyst

1. What is a loop invariant?

   A loop invariant is what a loop uses to continue for each iteration.

2. Explain how the loop invariant in power justifies correctness.

   (i < pow) tracks how many times the loop has run, hence tracking how many times the base has been multiplied by itself and ensuring that the output is correct.

3. Describe a situation where a for-loop would be more appropriate than a while-loop, and vice versa. Can you provide a code snippet for each scenario?

   A while loop was more appropriate in the case of our log function, whereas a for loop was more appropriate for our pow function

```
public long power(long input, long pow){
    long total = 1;
    for (int i = 0; i < pow; i++){
        total = multiply(total, input);
    }
    return total;
}

public long log(long base, long input){
    long total = 0;
    while (power(base, total) < input) {
        total++;
    }
    return total;
}
```

4. Explain how using methods can make a program easier to understand and maintain.

   Methods separate the code into pieces, making it so that the program is modular and each individual part can be maintained in isolation from the others.

5. Describe what a method signature is in Java. How does it help in method overloading?

   A method signature is a combination of the method’s unique name in combination with the method’s list of parameters. This method signature makes it possible to use the “same” method in multiple different contexts.

6. Consider a problem using nested loops. Explain how the control variables in the outer and inner loops interact with each other. Could you change the order of these loops without affecting the results? Why or why not?

   When nesting loops, the innermost code executes a number of times equal to the outer control variable multiplied by the inner control variable. These can, in theory, be swapped and have the result remain unchanging, but only if the control variables are unused in the code.