Report 3

1. Reflection

In this assessment, I learned and practiced a lot of object-oriented programming concepts such as inheritance, interfaces, abstract class, polymorphism, SOLID Principles of OOD, prototyping, also some knowledge on switch statements, unittests, UML diagram.

2. How did this design incorporate future growth?

This is my first system design project start from scratch, guided by SOLID Principles, from UML design, write a lot of tests and comments, create global class, abstract class, interface, and inherence and implements from them and build a whole system. I think this might be very similar as the real-world project in the workplace. I would definitly benefit from the habits formed by this project in the future groth, such as make a clear structure, design UML before coding, test driven and give lots of comments to increase readability.

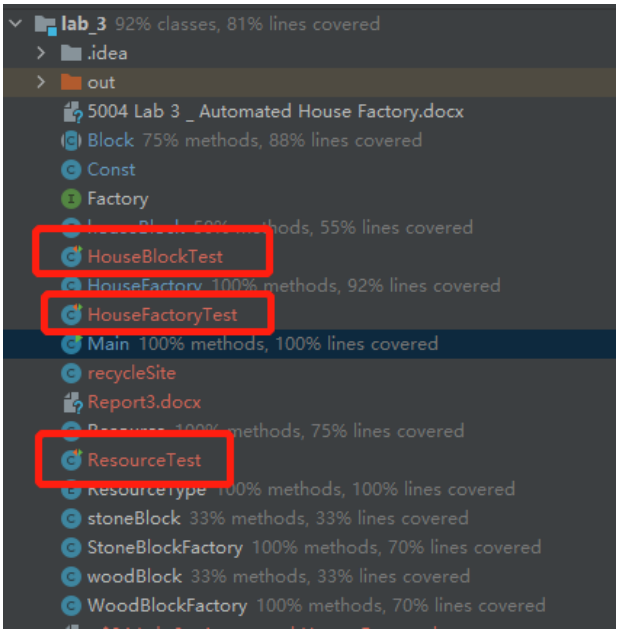
3. Did you change up the driver at all? If so, how?

I did not change the original driver since it works fine with my code.

4. Extensions

- **Create a JUnit test file for at least some of your classes**

  I have created 3 additional unittest classes, including HousefactoryTest, HouseblockTest, ResourceTest classes.



- **Add useful comments to the provided driver**

  I have added detailed comments on the provided driver

```java
/**
 * Main class for Automated House Factory Assignment.
 *
 * This class runs an infinite loop that generates a random resource,
 * then send it to a StoneBlockFactory or WoodBlockFactory to produce blocks,
 * then blocks are taken by the HouseFactory and a house is built
 */

Shawnsuun *
public class Main
{
    Shawnsuun *
    public static void main(String[] args) throws InterruptedException {
        // Declare a resource object for passing to the factories
        Resource resource;

        // Creating instances of three different factories
        Factory stoneBlockProducer = new StoneBlockFactory();
```

- **Find a way to exercise the break block function**

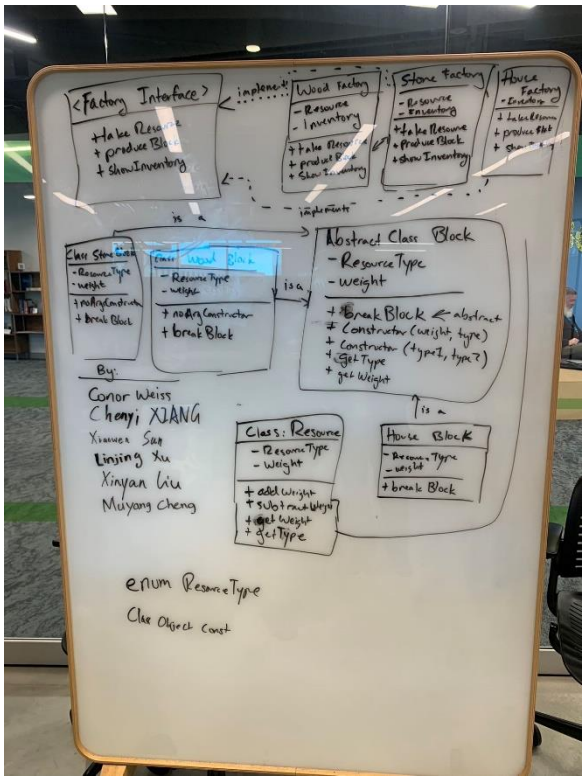
  I have created a recycleSite class to use the break block function in the recycle function

```java
public void recycle(Object block) {
    try {
        if (block == null) {
            throw new IllegalArgumentException("object is null");
        }
        //throw an exception if we send it something that isn't a resource
        if (!(block instanceof Block)) {
            throw new IllegalArgumentException("object is not a valid block");
        }
        switch (((Block)block).getType()) {
            case WOOD:
                this.woodBin.addAmount(((woodBlock) block).breakBlock().getAmount());
                break;
            case STONE:
                this.stoneBin.addAmount(((stoneBlock) block).breakBlock().getAmount()
                break;
            case HOUSE:
                this.stoneBin.addAmount(((houseBlock) block).breakBlock().getAmount()
                break;
        }
    } catch (IllegalArgumentException ex) {
        System.out.println(ex);
```

- **Add more than the requested exception handling**

  Several exception handling are added such as the above recycle function in recycleSite class.

5. recitation UML diagram



6. Grading Statement

The project fulfills all basic functions required by the lab instructions, there is 89 points on it.
Extensions: 4 extensions are implemented for 10 points.
I wrote a lot of tests and comments and this might add points in the Creative or went above and beyond part.
For the above reasons, I think this work should be graded as 99 or above.