COMP110 - Worksheet 03 Name: _____ PID: _____

Please print out and fill in this worksheet by hand. Make sure that when submitting your assignment to Gradescope, you scan each page as a whole page, and that they are uploaded in the correct order, in the proper orientation, with no extra pages included. **Points will be deducted for not following these guidelines.**

1. In the space below, create a class called `Triangle` with the following specifications:

1. This class contains 3 side length properties of type `number` (named `a`, `b`, and `c`), and a `string` property named `type`.
2. Write a constructor for the `Triangle` class that takes in three `number`s which will become the values of `a`, `b`, and `c`, respectively. The constructor should also set the value of `type` by calling the method described in part 3, below.
3. Write a method called `determineType` that determines if a given `Triangle` is scalene, isosceles, or equilateral based on its side length values, and that returns the correct `Triangle` type as a `string`.

Example:

```
let aTriangle = new Triangle(1, 2, 2); //aTriangle.type is "isosceles"
```

2. In the space below, create a class called `Donut` with the following specifications:

1. Each `Donut` has a `boolean` property called `hasIcing`, a `string` property called `flavor`, and two `number` properties called `sprinkleCount` and `sweetness`.
2. The class should have a constructor that takes in and sets up each of its property's values.
3. Write a method called `sprinkleShaker` that takes in a `number`, increases the `sprinkleCount` by that amount, and increases the `sweetness` by that amount divided by 3.
4. Write a method called `calorieCounter` that returns a `number`. If the `flavor` of the `Donut` is "maple" or "lemon," increase the calorie count by 30, otherwise increase it by 45. If the `Donut` has icing, increase the calorie count by 100. Finally, increase the calorie count by half the number of sprinkles. Note that the calorie count is not a property of a `Donut`, but a value that should be calculated and returned only when this method is called.

3. Given the code listing below, draw an environment diagram then answer the questions that follow. In each frame on the stack, remember to include a space for the return address and return value, if any.

```
1  class Point {
2      x: number;
3      y: number;
4
5      constructor(x: number, y: number) {
6          this.x = x;
7          this.y = y;
8      }
9  }
10
11 class Line {
12     start: Point;
13     end: Point;
14
15     constructor(one: Point, two: Point) {
16         this.start = one;
17         this.end = two;
18     }
19
20     stretch(amt: number): void {
21         this.start.x -= amt;
22         this.end.x += amt;
23     }
24
25     translate(): void {
26         this.start.x += this.end.y;
27         this.end.x = 5 + this.start.x;
28         this.start.y = this.end.x % this.start.y;
29         this.end.y--;
30     }
31 }
32
33 export let main = async () => {
34     let p1 = new Point(3, 1);
35     let p2 = new Point(4, 8);
36     let initial = new Line(p1, p2);
37     let final = initial;
38     final.translate();
39     final.stretch(2);
40     print(initial.start.x + "␣" + initial.start.y);
41     print(final.start.x + "␣" + final.start.y);
42     // Pause Here
43 };
44
45 main();
```

The following questions are about the state of the program at the moment it is paused. If a question is asked about a name that has not been initialized, then respond with `undefined`.

3.1 How many objects are on the heap?

3.2 From `main`'s frame, use name resolution to look up the value of `initial.start.x`. What is its value?

3.3 From the final stack frame, use name resolution to look up the value of `final.start.x`. What is its value?

3.4 From the final stack frame, use name resolution to look up the value of `this.start.y`. What is its value?

3.5 What is the printed output of this program?

4. Given the code listing below, draw an environment diagram then answer the questions that follow. In each frame on the stack, remember to include a space for the return address and return value, if any.

```
1  class Chair {
2      stiff: boolean = false;
3      blanket: boolean = false;
4      book: boolean = false;
5      coziness: string = "TBD";
6
7     constructor(s: boolean, bla: boolean, boo:
           boolean) {
8         this.stiff = s;
9         this.blanket = bla;
10        this.book = boo;
11    }
12
13    cozyLvl(): void {
14        if ((this.stiff) && (!this.blanket && !this.
              book)) {
15            this.coziness = "poor";
16        } else if ((this.blanket && this.book) && !
              this.stiff) {
17            this.coziness = "excellent";
18        } else {
19            this.coziness = "average";
20        }
21    }
22
23    mix(a: boolean): void {
24        this.blanket = !this.stiff;
25        this.stiff = !a;
26        this.book = (this.blanket && a);
27    }
28 }
29
30 export let main = async() => {
31    let mystery = new Chair(false, true, false);
32    mystery.cozyLvl();
33    print(mystery.coziness);
34    mystery.mix(true);
35    mystery.cozyLvl();
36    print(mystery.coziness);
37    let alias = mystery;
38    alias.stiff = !alias.stiff;
39    alias.blanket = mystery.book;
40    alias.book = false;
41    // Pause Here
42 };
43
44 main();
```

The following questions are about the state of the program at the moment it is paused. If a question is asked about a name that has not been initialized, then respond with `undefined`.

4.1 How many `Chair` objects are on the heap?

4.2 What is the value of `mystery.stiff`?

4.3 If you called `cozyLvl()` on `alias`, what would its `coziness` value be?

4.4 Create a `Chair` object below. The method `cozyLvl` would change this object's `coziness` property to `"poor"`.

4.5 What is the printed output after the `main` function completes?

5. Given the definition of the class `Foo`, fill in the blanks with the variable type, return type, property name, or keyword that would fit in each numbered blank in the code listing.

```
1  class Foo {
2     numbers: number[] = [];
3     words: string[] = [];
4
5     constructor(n: number[], s: string[]) {
6        ____1____.numbers = n;
7        this.____2____ = s;
8     }
9
10    a(i: ____3____): ____4____ {
11       return this.numbers[i] + this.words[i];
12    }
13
14    b(i: ____5____): ____6____ {
15       let arr = [];
16       for (let j=0; j < this.numbers.length; j++) {
17          arr[i] = "" + this.numbers[i] + "";
18       }
19       return arr;
20    }
21
22    c(i: ____7____, j: ____8____): ____9____ {
23       this.numbers[j] = i.numbers[j];
24    }
25
26    d(i: ____10____): ____11____ {
27       return this.words[i].length;
28    }
29
30    e(j: ____12____): ____13____{
31       let i: ____14____ = [];
32       for(let k = 0; k < i.length; k++) {
33          i[j] = "0";
34       }
35       let foo = new ____15____([1,2,3], ["aye", "
             bee", "cee"]);
36       foo.____16____ = i;
37       return foo;
38    }
39 }
```

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____
15. _____
16. _____

6. Given the code listing below, answer the questions that follow.

```
class Trainer {
    name: string;
    x: number;
    y: number;
    facing: number;

    constructor(name: string) {
        this.name = name;
        this.x = 1;
        this.y = 2;
        this.facing = 3;
    }

    stepForward(): void {
        if (this.facing === 0) {
          this.x++;
        } else if (this.facing === 1) {
          this.y++;
        } else if (this.facing === 2) {
          this.x--;
        } else {
          this.y--;
        }
    }

    turnRight(): void {
        if (this.facing < 3) {
          this.facing++;
        } else {
          this.facing = 0;
        }
    }

    info(): void {
        let s = this.name;
        s += "␣is␣at␣";
        s += this.x + ",␣" + this.y;
        print(s);
    }
}
```

6.1 Given the following code listing, what is the printed output? If there is no printed output, write "N/A".

```
let trainer: Trainer = new Trainer("Luke");
trainer.stepForward();
trainer.turnRight();
trainer.stepForward();
trainer.info();
```

6.2 Implement the method `runForward` such that it will call its `Trainer`'s `stepForward` method `n` times. The `runForward` method should take in the value of `n`.

6.3 Declare and initialize a new `Trainer` object. Then make use of the methods in the `Trainer` class by calling them on your newly created `Trainer` object so that calling `info()` will output `"<name> is at 111, 1"`. You may use the `runForward` method that you defined from the previous question.