11/17: Module 26

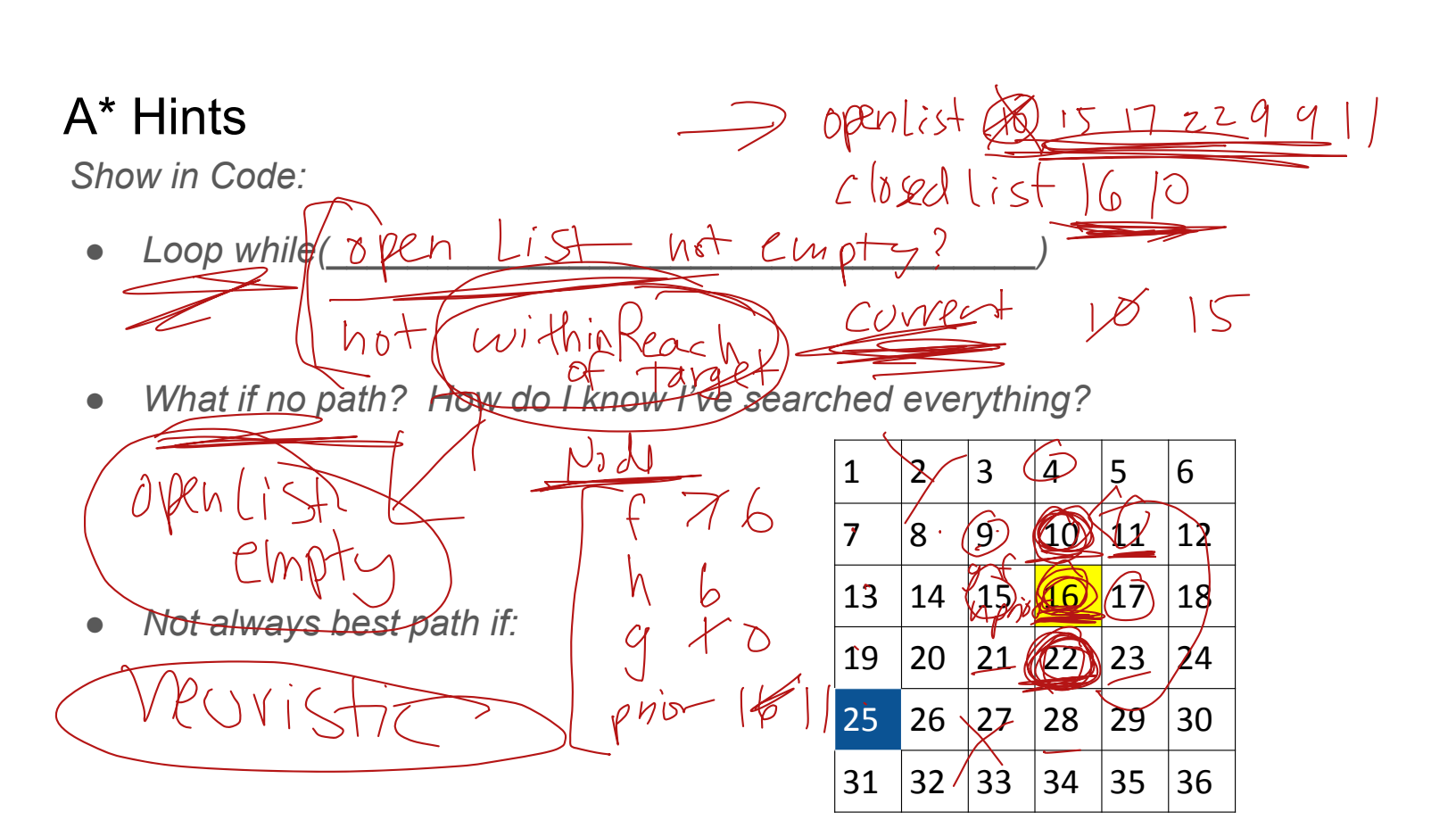
A* Hints, Polymorphism Wksts, Final Study,
P5 Intro (Pick partner potentially)

A* Implementation Hints

A* Hints

Show in Code:

- *Loop while(* open List not empty? *)*
- *What if no path? How do I know I've searched everything?*
- *Not always best path if:*



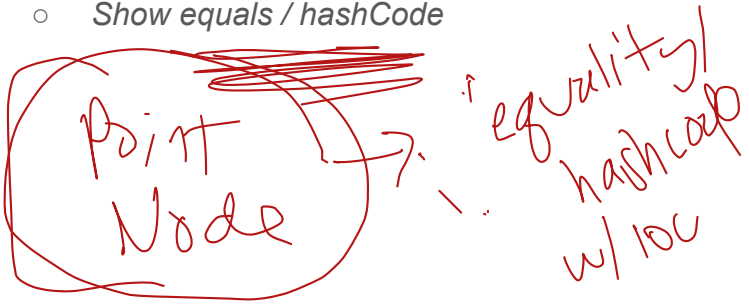
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

A* Hints

Show in Code:

- *How to build path?*
  - *Closed list is NOT path*

- *How to uniquely identify node?*
  - *Show equals / hashCode*



1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

A* Algorithm: the Algorithm

A* Algorithm

1. Choose/know starting and ending points of the path
2. Add start node to the open list and mark it as the current node
3. Analyze all valid adjacent nodes that are not on the closed list
    a. Add to Open List if not already in it
    b. Determine distance from start node (g value)
        i. g = Distance of current node from start + distance from current node to adjacent node
    c. If the calculated g value is better than a previously calculated g value (or if there was no previously calculated g-value), replace the old g value with the new one (or add the g-value) and:
        i. estimate distance of adjacent node to the end point (h)
            1. This is called the heuristic. It can be Euclidean distance, Manhattan distance, etc.
        ii. Add g and h to get an f value
        iii. Mark the neighbor node's prior vertex as the current node
4. Move the current node to the closed list
5. Choose a node from the open list with the smallest f value and make it the current node
6. Go to step 3

Repeat until a path to the end is found.

Project 5 Intro

Overview

- Show instructions

- Can pick a partner:
  - Will need to copy one of your code into a new private github repo to share
  - THEN one of you can copy and paste back into the Github Classroom repo
  - The other partner must upload a WORLD_EVENT.txt saying who they worked with and where to code is

- Can start:
  - Planning
  - Finding / generating bmp files

Polymorphism Practice

Polymorphism 1

Finish reviewing

Recall:

- Reference variable type
- Compiler uses
- Actual type in memory
- Instiated object

```
Food f = new Candy();
```

Practicing Polymorphism

Combining:

- Overloading
- Overriding

Recall:

- Compiler only uses reference variable type (param or declared var)
  - `Object o`
  - `List<Dog> l = new ArrayList<>();`
  - `Entity e = new Stump(...);`

- Compiler will choose by:
  - Actual type
  - Closest is-a relationship (can cast UP)
  - Convert w / autoboxing

- At runtime, the actual object's method is called (if overridden)

Practicing Polymorphism

If overloading AND overriding, still:

1) Be sure to pick method signature based on ref var type
2) THEN check if the overloaded method the compiler picked was overridden

Try Polymorphism pt 2

Review

Final Quiz

Study Material

- Show where posted
- Exceptions not included on practice quiz yet, but will give separate practice problems for

Assignments

Lab 8:

- Due Monday week 10

Project 4:

- Friday (110)
- Monday week 10 (100)

Final:

- Wed of week 10

Final Project:

- Wed 12/8 (8am)
- Can work in pairs