

Activity Diagrams

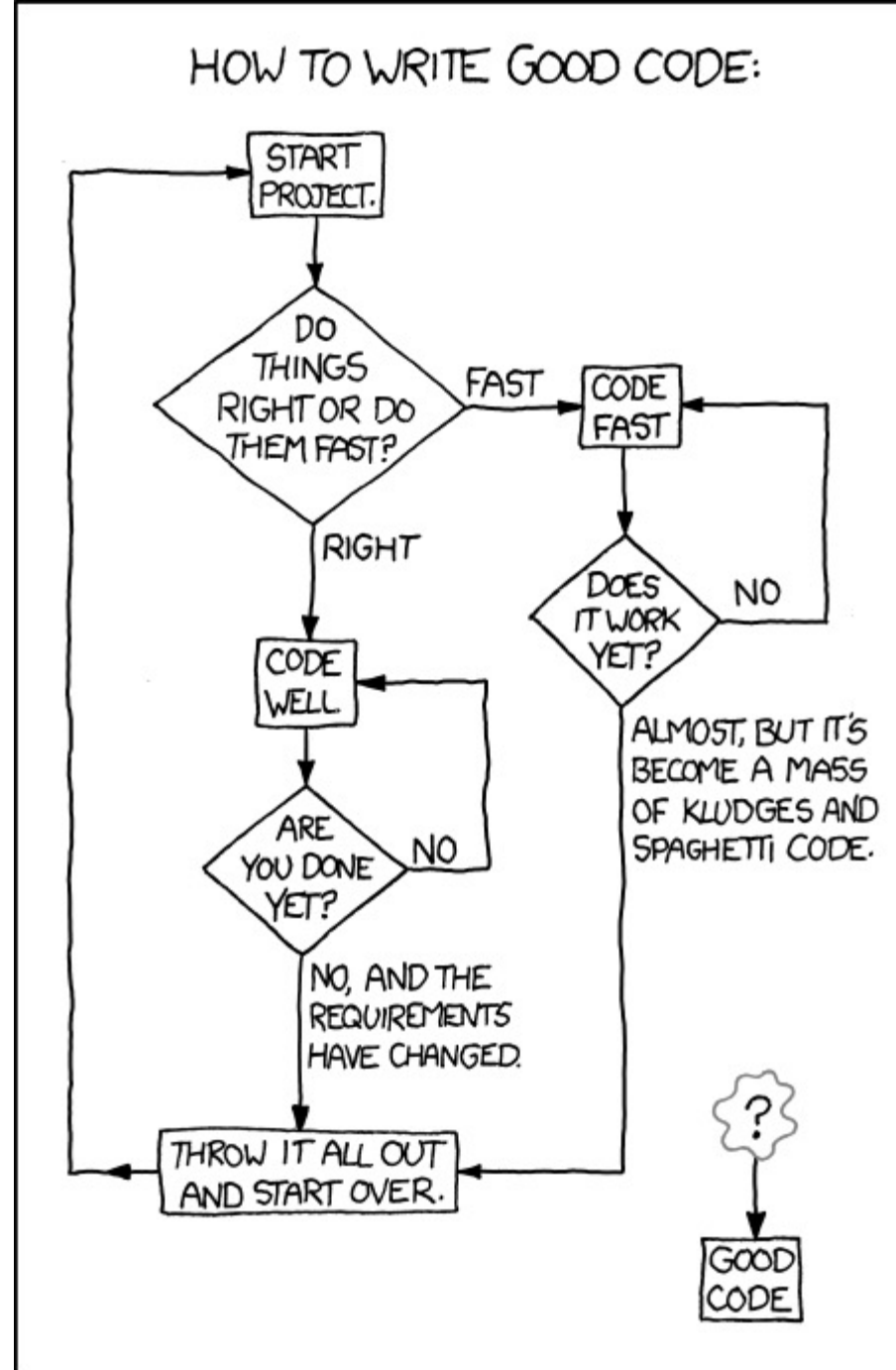
CPSC 2150

Activity Diagrams

- Activity diagrams are a UML diagram we will make in this course
- Similar to a flow chart
 - They show the logical flow of a program
- Can be drawn to help explain existing code
 - Map out the logical flow
 - Identify test cases
- Can be drawn to help design your code
 - Make an activity diagram of each function
 - Work out the decisions and the logic before you start writing the code

General Flow Chart

- Diamonds split our flow up based on the answer to a question
- Rectangles are activities
- UML has a stricter notation



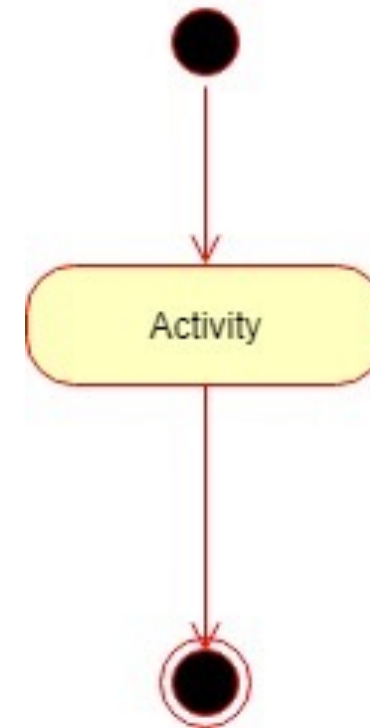
UML Activity diagram

- A solid black circle is the start of our activity diagram
 - The beginning of a function
- A solid black circle with another circle around it is the end
 - Where we return from the function
- **Note:** Diagrams.net (Formally Draw.io) added color



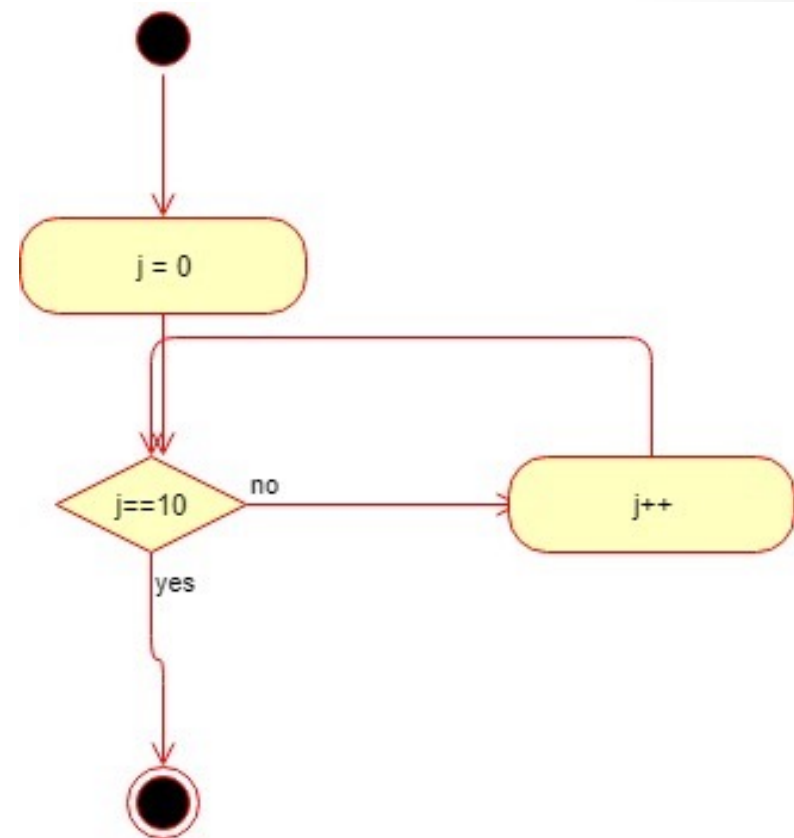
Activity

- In a UML activity diagram, activities are rounded rectangles
 - Often ovals when hand drawn
- An activity can be one or more lines of code that don't have any decisions or loops in them summarized as one task
 - “Solve for monthly payment”
 - “Divide by 4”
 - Calling another method
 - The method itself may have some logic in it, but we'll often make a separate activity diagram for it



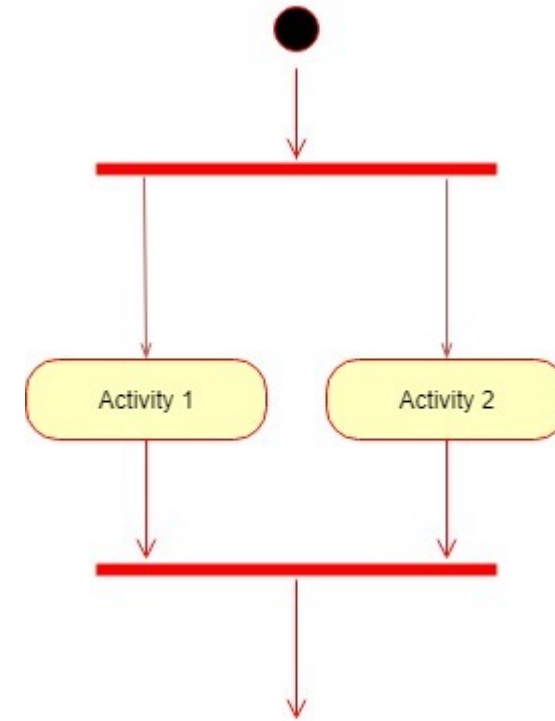
Decisions

- Decisions are represented as a diamond
 - An arrow flows in
 - The question or logical statement is listed above or in the diamond
 - 2 arrows flow out, one yes, and one no
 - It's tempting to add multiple options, but it will match our code better if we stick to Boolean true/false or yes/no decisions
- Loops can be handled by having the “no” path return to the decision
 - Usually after some activity

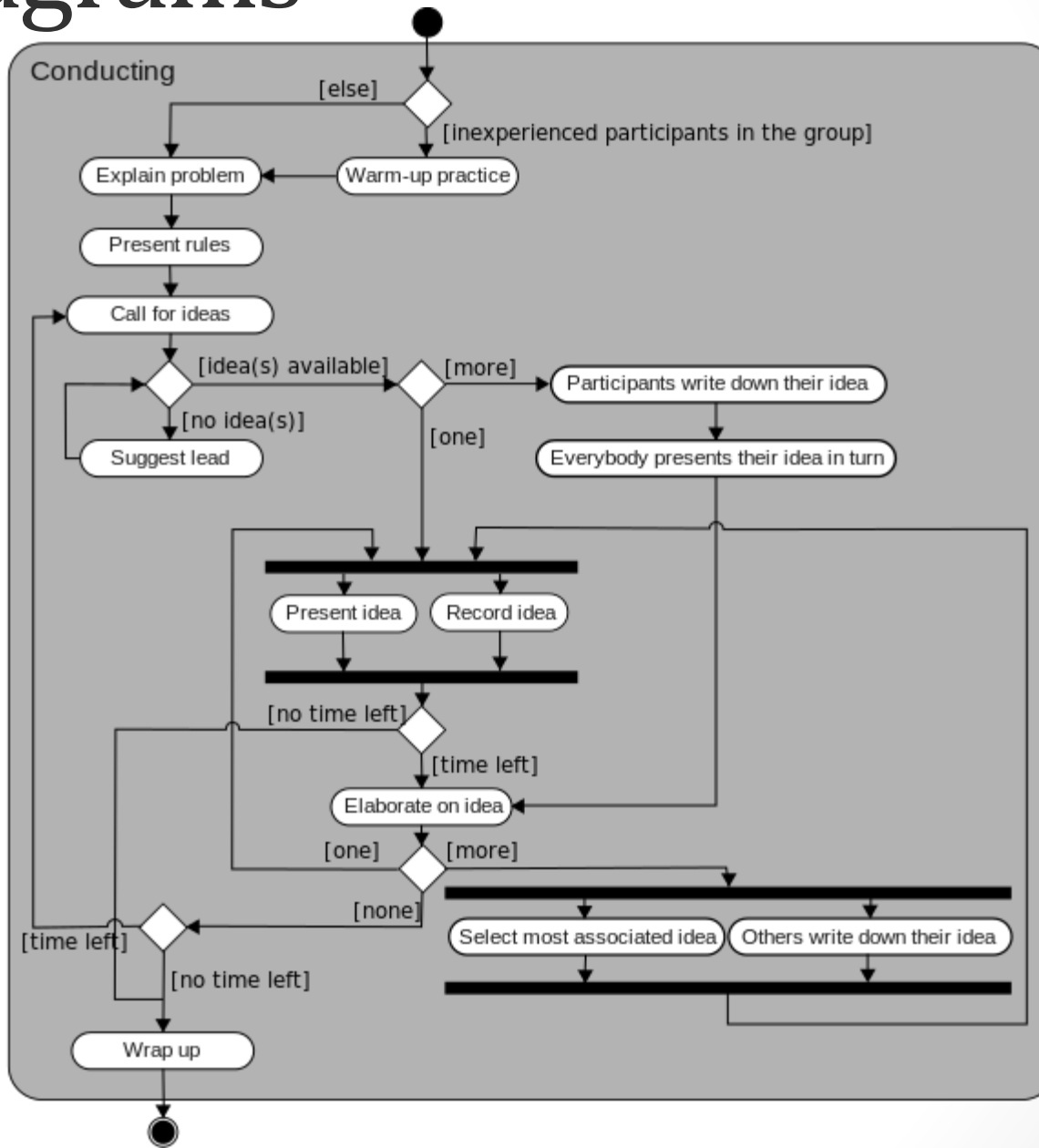


Concurrent Activities

- You may have activities that can happen at the same time
 - Can happen in any order
 - Even if we aren't going to write our program to use parallel processing, we still want to show that it is possible in our diagram
- Use a bar to show the paths splitting, and another bar to rejoin them
 - Can also use the bar to rejoin paths that split due to a decision
 - Or have them rejoin by going to the same activity



UML Activity Diagram



Benefits

- You can work out how to solve the problem at a high level before you worry about individual steps and syntax
 - Individual steps are activities, we'll fill in the details later
- Helps identify the general logic of your program
 - Then you can make you loops and if statements in the code and fill in the details
 - How many times is the error in the code because you end the block of a loop too early or too late?
- Have an easy to reference chart to help you
 - Pick test cases that guarantee 100% code coverage
 - Explain your code to other developers on your team

The end

- In this course you will have to draw activity diagrams for every method in your code
 - This will help you design your code
- Again, it may not seem necessary now, but it will help as your programs get more advanced