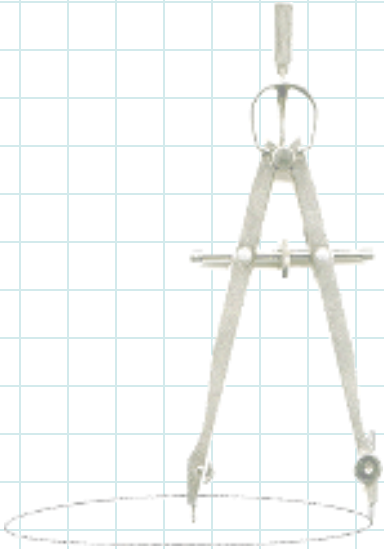
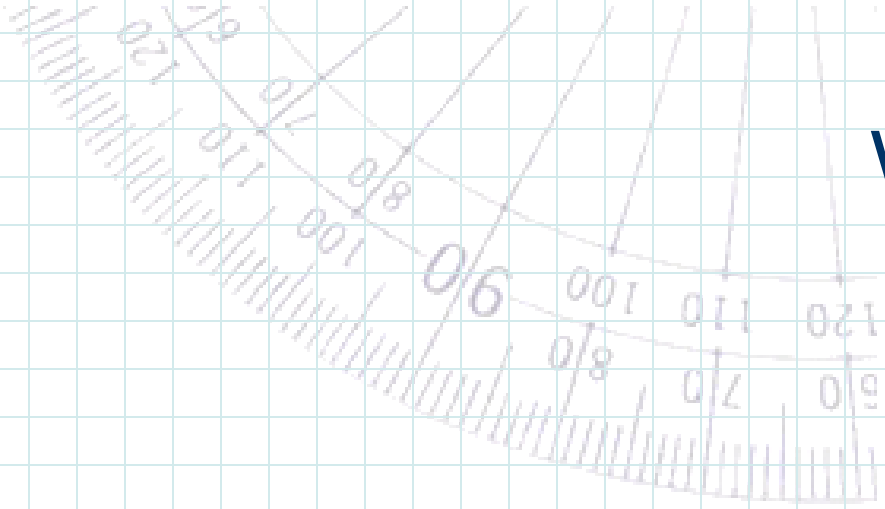
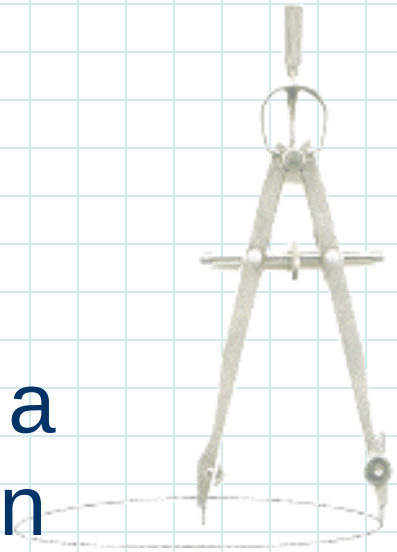
Decision Statements



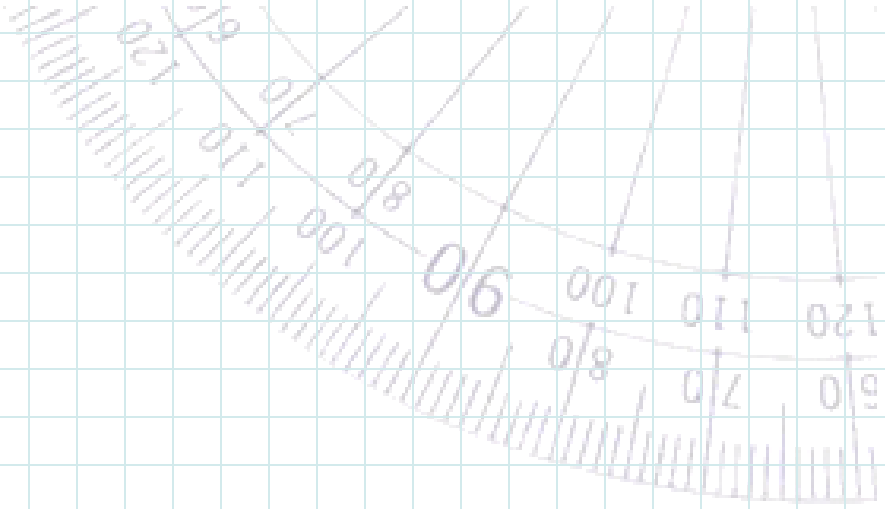


Website Production

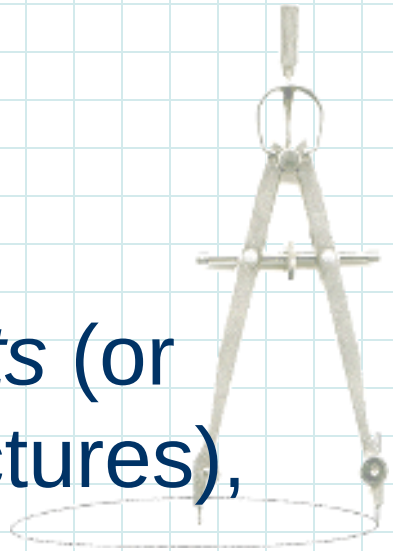
Q: What is a decision?



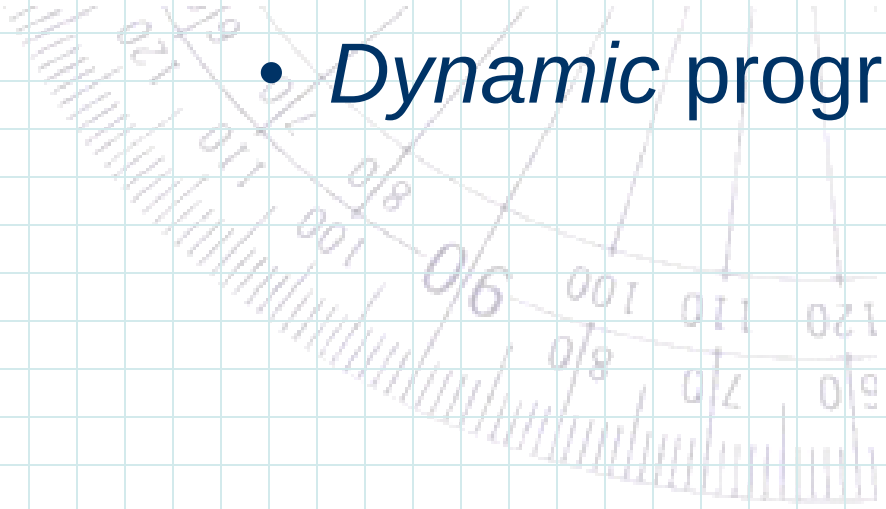
- Something that represents a branching point in a solution
- Outcomes are often dependent on initial conditions



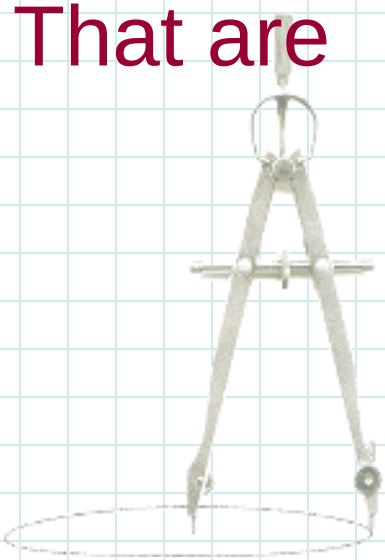
Decisions in Programs



- Without *decision statements* (or other dynamic control structures), programs are static
- *Static* programs do exactly the same things each time they are executed
- *Dynamic* programs do not



Decisions are Based On Statements That are True or False



- Boolean Algebra or Logic

  with operations and, or and not
- Relational Operations

  with is less than(<)

  is greater than(>)

  is equal(==)

  is less than or equal to(<=)

  is greater than or equal to(>=)

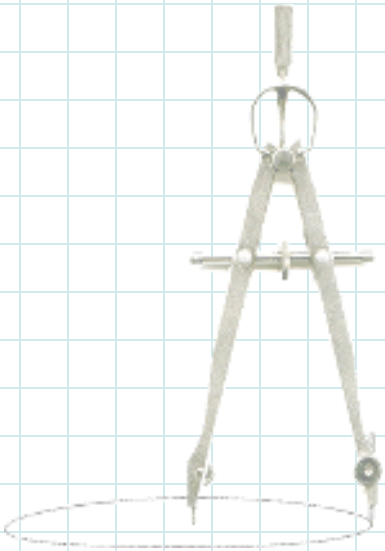
  is less than or greater than(<>)

  or is not equal to(<>)

Boolean Algebra

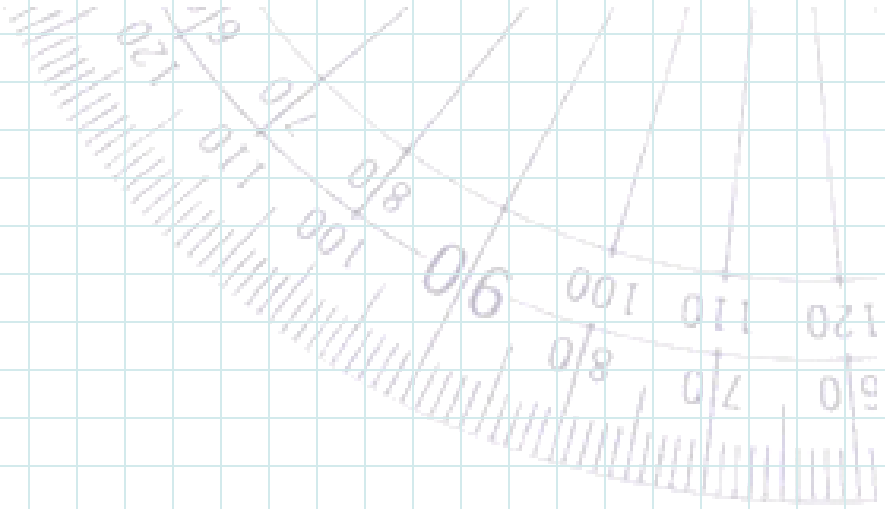
- Based on values that are either *True* or *False*
- True and False values are often represented by *1's* and *0's*, respectively
- Operations are And, Or and Not

Boolean Algebra
Logical Operation: And

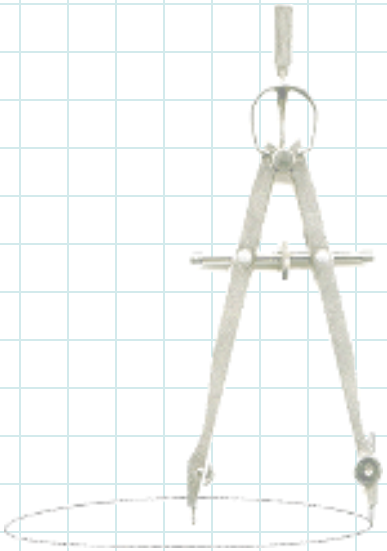


- $A \wedge B$ *(A and B)*
- Expression is True *if and only if A* and *B* are both true

A\B	T	F
T	T	F
F	F	F



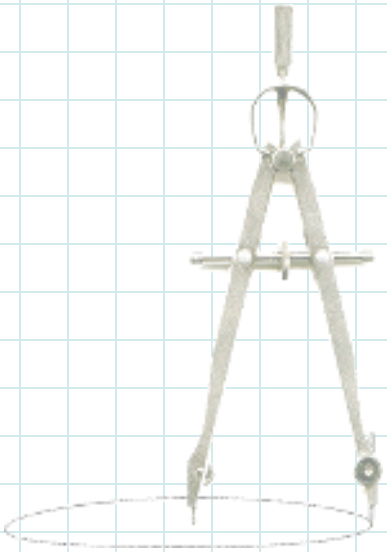
Boolean Algebra
Logical Operation: `Or`



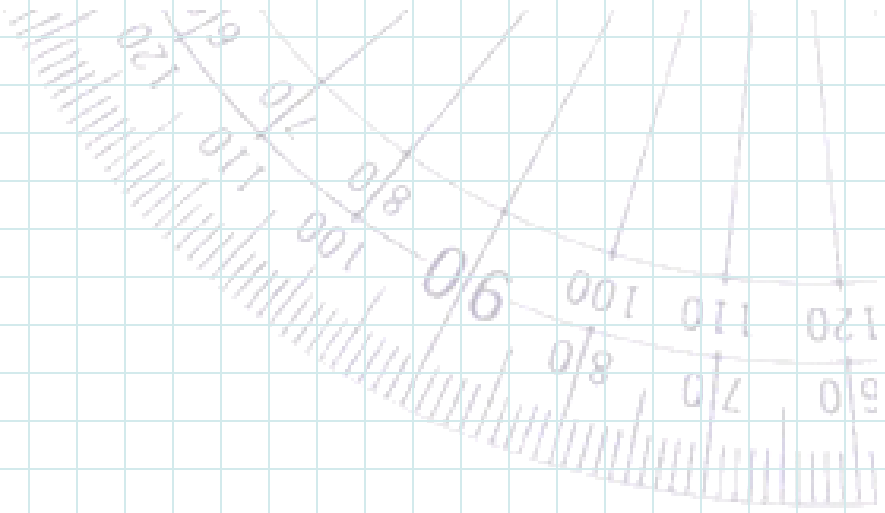
- $A \vee B$ *(A or B)*
- Expression is True if either *A* or *B* are True
- Note: Also True when *A* and *B* are both True

A\B	T	F
T	T	T
F	T	F

Boolean Algebra
Logical Operation: `Not`

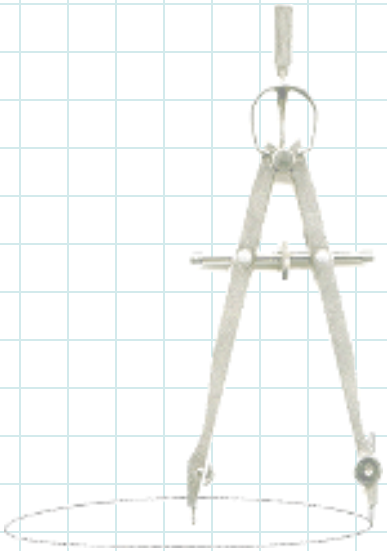


- ¬*A* *(not A)*
- Expression returns the negation of A



A	¬A
T	F
F	T

Logical Operations: *Exercises*

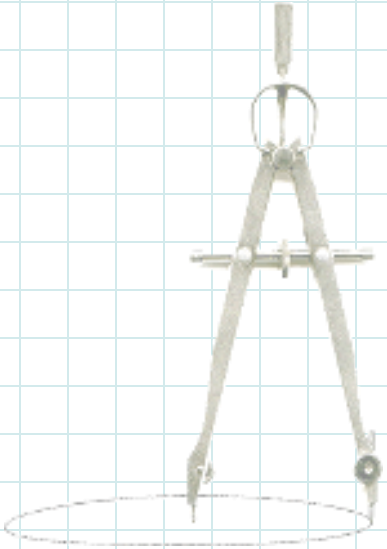


A = True, B = True, C = False

1. $A \vee B$
2. $A \wedge C$
3. $A \vee B \wedge C$
4. $(A \wedge B) \vee (A \wedge C)$
5. not A

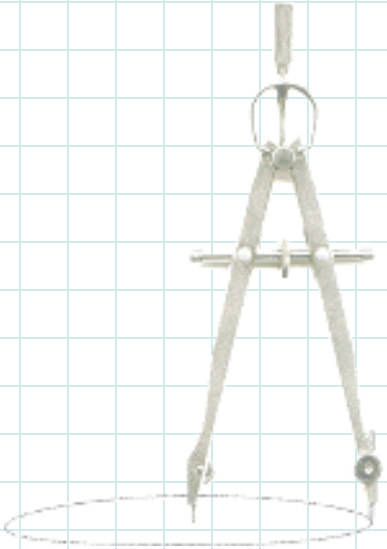
See next slide for answers

Answers to Logical Operations Problems



1. True
2. False
3. False
4. True
5. False

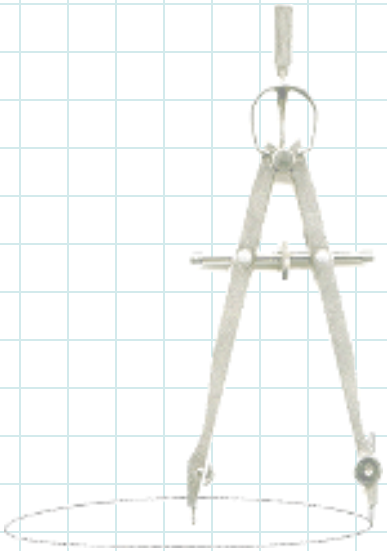
Relational Operations



- $A < B$ “A less than B”
- $A > B$ “A greater than B”
- $A = B$ “A equal to B”
- $A \leq B$ “A less than or equal to B”
  “A not greater than B”
- $A \geq B$ “A greater than or equal to B”
  “A not less than B”
- $!(A = B)$ “A not equal to B”

(or A<>B) “A less than or greater than B”

Relational Operations: *Exercises*

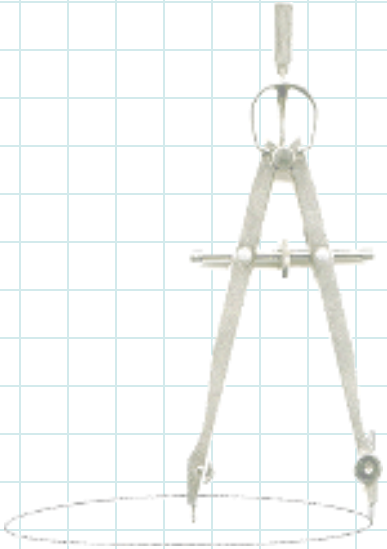


$A = 5, B = 3, C = -7$

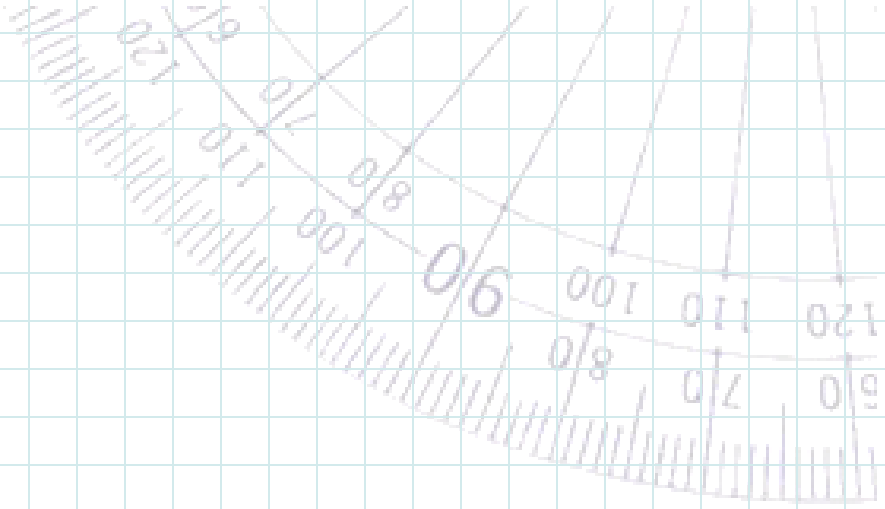
1. $A < B$
2. $A \geq C$
3. $(A < C) \vee (B < C)$
4. !(A < B)

See answers on next slide

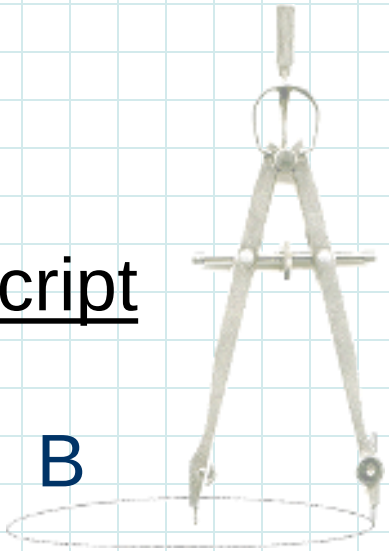
Answers to Relational Operators Problems



1. False
2. True
3. False
4. True



Boolean or Logical Operations



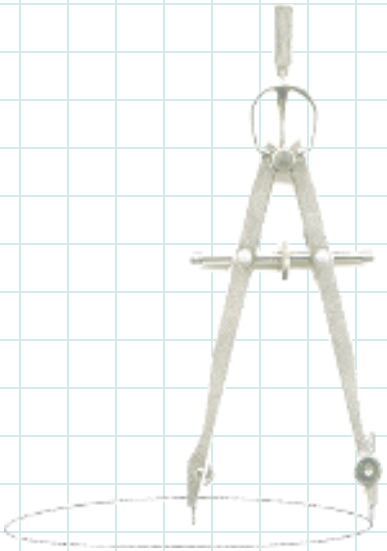
Traditional		Javascript
$A \wedge B$	*and*	A && B
$A \vee B$	*or*	A || B
$A < B$	*is less than*	A < B
$A > B$	*is greater than*	A > B
$A = B$	*is equal to*	A = = B
$A \geq B$	*is greater or equal to*	A > = B
$A \leq B$	*is less than or equal to*	A < = B
$A \neq B$	*is not equal to*	A < > B

Try this! There's a form with JavaScript that can solve this problem in movies.html

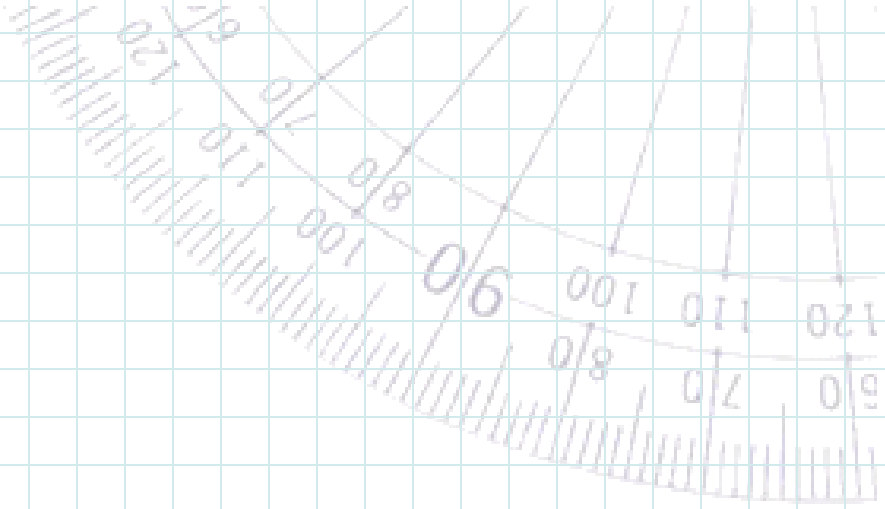
Problem:

- You'd like to go see a movie.
- The movie costs $8.00, a soda costs $2.50 and a large popcorn costs $4.50.
- Based on the amount of money in your pocket, determine whether you could...
  (a) See the movie and buy a soda,
  (b) See the movie, and buy soda and popcorn, or
  (c) Stay home

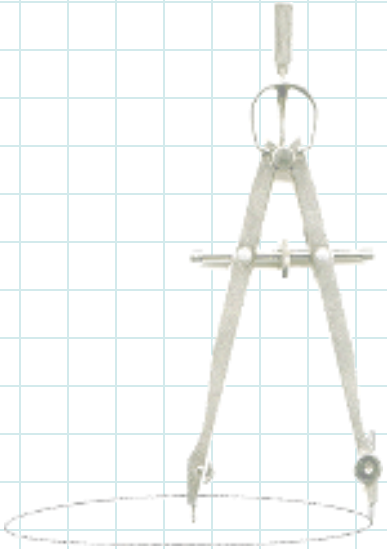
Problem Solving with Know, Need, Do Method



Method by Professor WJoel



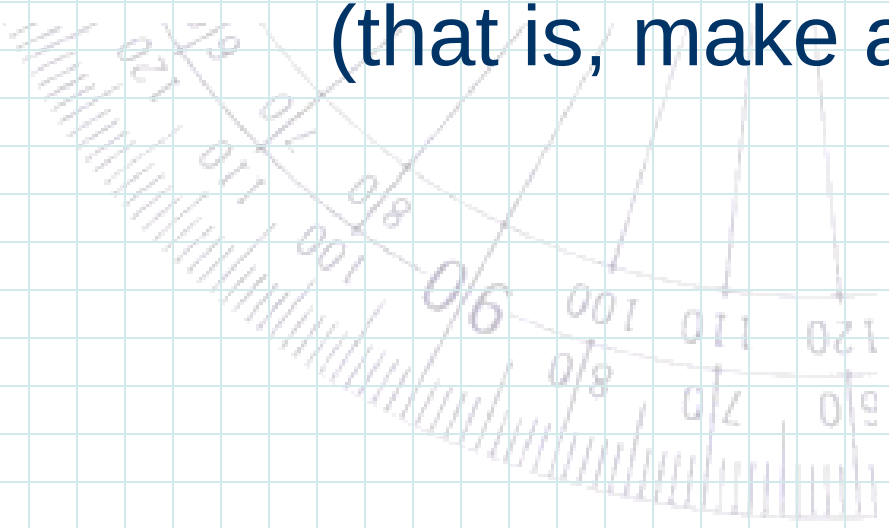
Know?



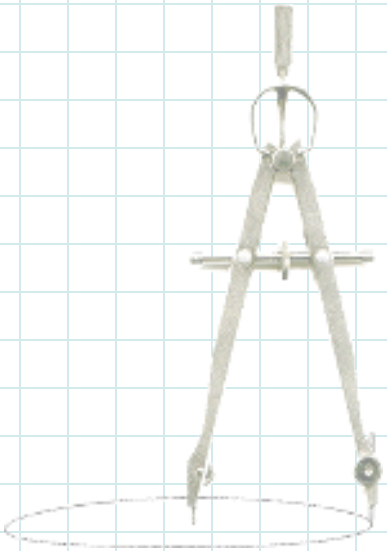
- Movie costs $8.00
- Soda costs $2.50
- Popcorn costs $4.50
- How much money I have in my pocket

Need?

- Cost of movie and soda
- Cost of movie, soda and popcorn
- Way to select one of the three options
  (that is, make a decision!)



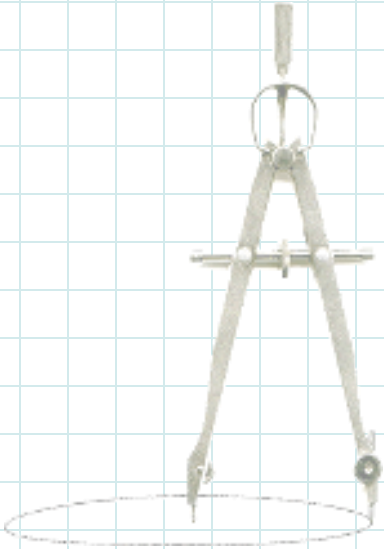
Do?



- Option (a) costs $10.50
- Option (b) costs $15.00
- Option (c) costs nothing
- What next?

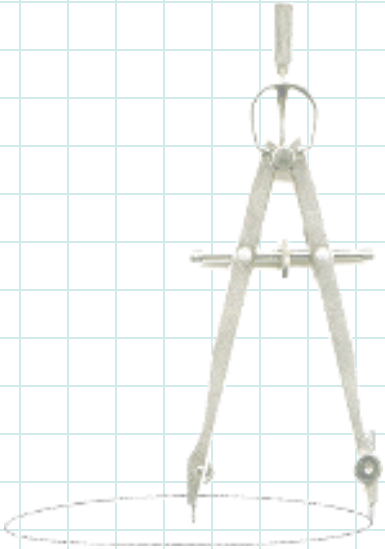
How about a diagram?

- This is called a *flowchart*



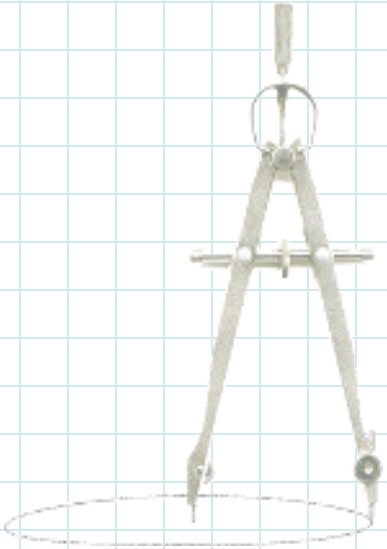
How about a diagram?

- Boxes represent actions



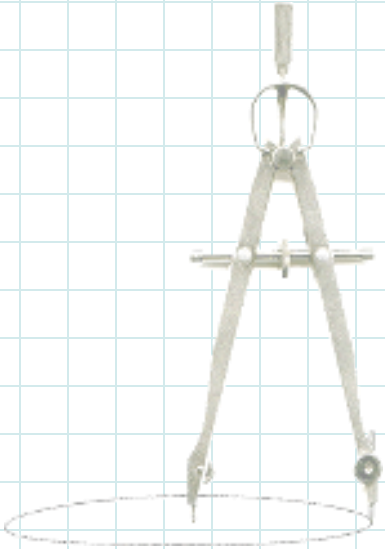
How about a diagram?

- Diamonds represent decision points



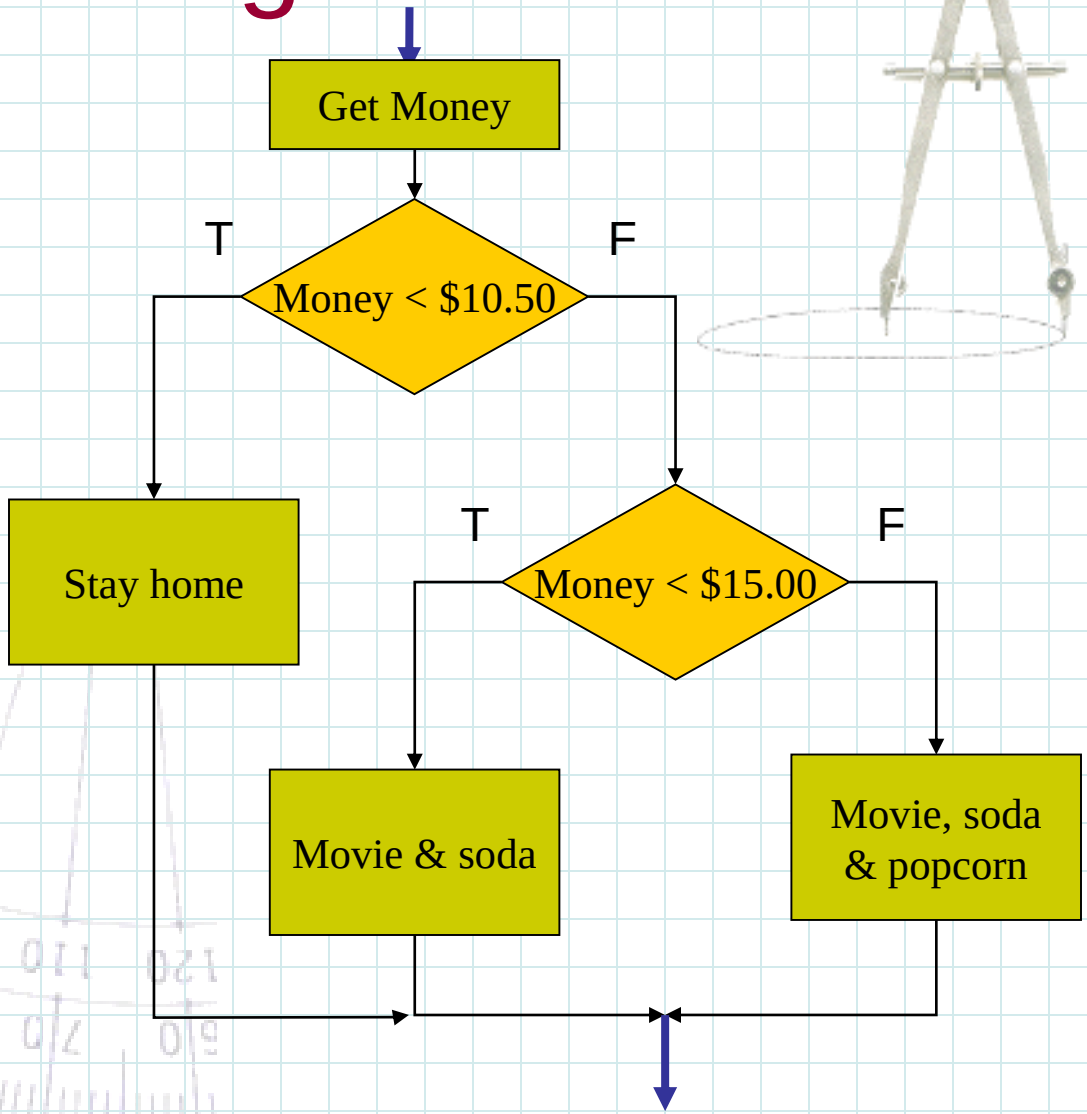
How about a diagram?

- Arrows show flow

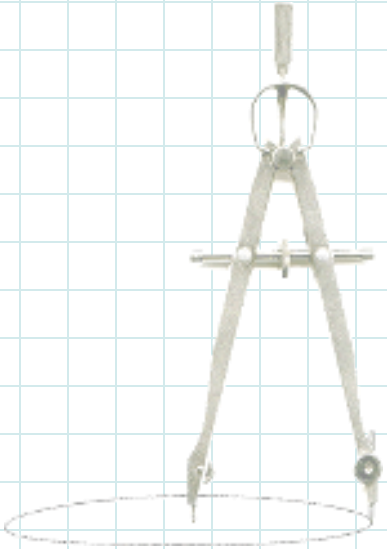


How about a diagram?

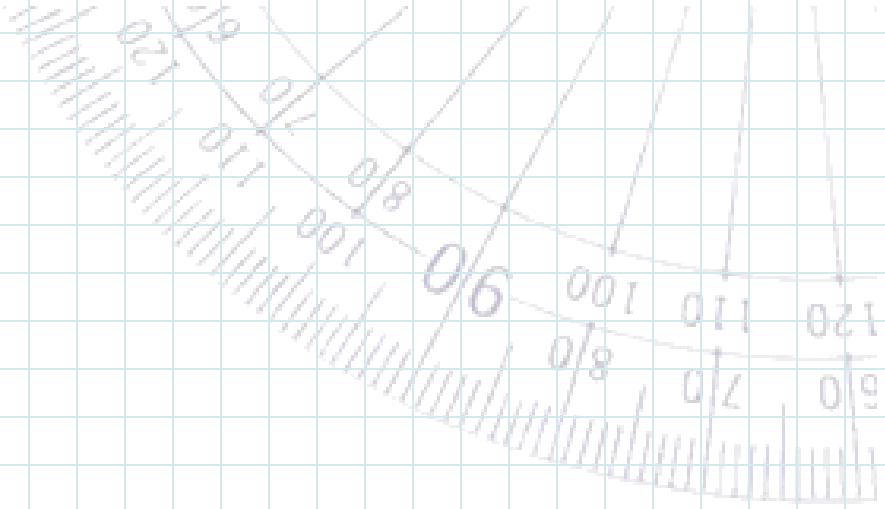
- The arrow at the top tells us there were previous steps
- The arrow at the bottom tells us there are subsequent steps



How would I write this?



- Using *Pseudocode*
- What is Pseudocode?

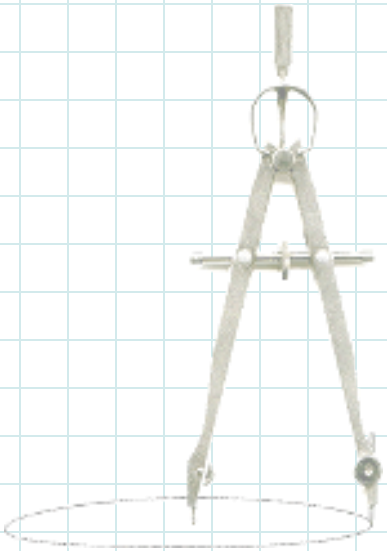


Pseudocode

- Looks like a programming language
- Has all the structure of a programming language
- Has a ***verrrrrry*** loose syntax

- Is just an outline for a set of instructions, possibly for a computer to follow

Pseudocode

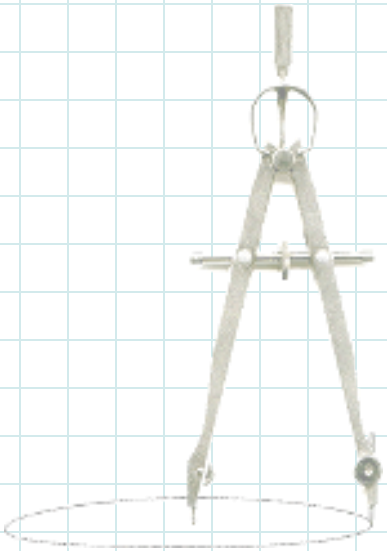


- Example:

```
get x
result ← x^2 + 5x + 7
print result
```

- That's it!

One more time!



- *Pseudocode...*

  If (Money < $10.50) then

  Stay home

  else If (Money < $15.00) then

  Movie, soda

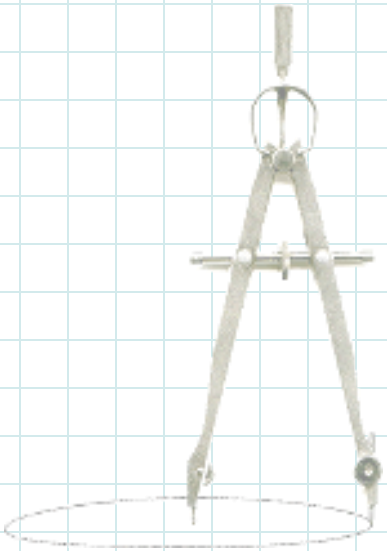
  else Movie, soda, popcorn

How would I write this?

- The problem statement *tells* us the individual costs
- No need to ask the user for them

- Problem:
- You'd like to go see a movie.
- The movie costs $8.00, a soda costs $2.50 and a large popcorn costs $4.50.
- Based on the amount of money in your pocket, determine whether you could...
  (a) See the movie and buy a soda,
  (b) See the movie, and buy soda and popcorn, or
  (c) Stay home

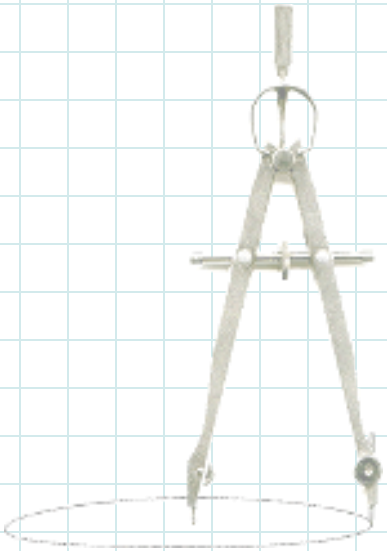
How would I write this?



The cost of each option is

- Movie: $8.00
- Movie & soda: $10.50
- All three: $15.00

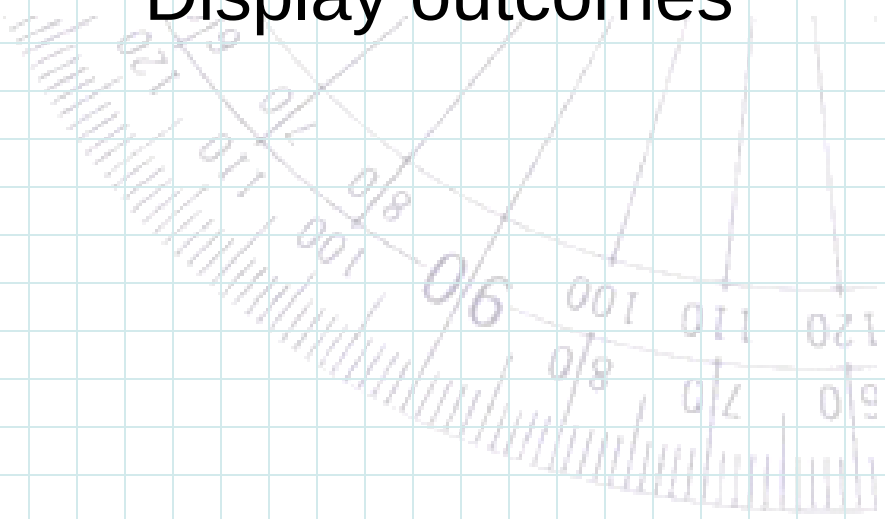
How would I write this in pseudocode?



Ask user for how much money she has

Decide outcomes

Display outcomes



How would I write this?

- Next, we need to make sure we have a complete algorithm, so we refine our pseudocode

```
Input Money
If (Money < $10.50) then
    Display “Stay home.”
    else If (Money < $15.00) then
            Display “Go to a movie;buy a
soda.”
         else
            Display “Go to a movie; buy a
                soda and popcorn.”
```

- Almost done!

In JavaScript (code): If Statement

```
if (condition)
    then-true-action
else
    else-false-action
```

Complex statement

```
if (a<5) {
  b = 1;
  c = 2;
  }
else
  b = 3;
```

Simple statement

Program

- Okay, after finishing this PowerPoint, run and view the webpage movies.html based on our pseudocoded algorithm and some form design, which uses a JavaScript script that makes a decision using if-then-else statements

Flowcharts

To follow a flowchart, start at the top and follow the arrows

When you come to an input or output box, get the necessary data for variables and write it down on a piece of paper. When you come to an assignment statement change the value on your paper

For a decision box, decide whether the expression is true or false and follow the appropriate arrow