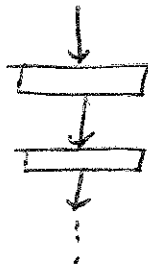
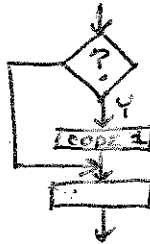


SELECTIONS

PROGRAM SEQUENCE

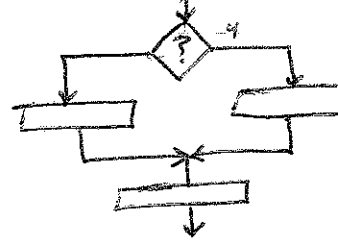


1-WAY SELECTION



if()

2-WAY SELECTION



if()-else

BOOLEAN VALUE

THE QUESTION MUST BE A YES-NO (true-false) TYPE

RELATIONAL OPERATORS

== < >
!= >= <=

```
if (a > b)
    System.out.println("a is greater THAN b");
else
    System.out.println("a is not greater than b");
```

boolean variable

```
boolean notDone;
```

```
notDone = true;
```

```
if(notDone)
    SYSTEM.out.println("Not Done Yet");
```

PITFALLS

- = vs ==
- DANGLING ELSE
- FLOATING POINT EQUALITY
- SEMICOLON AFTER if()

LOGICAL OPERATORS

! not; && and; || or; ^ exclusive-or

&& and || ARE SHORT-CIRCUITING (conditional-AND)
& and ! ARE UNCONDITIONAL.

AMFAD

RANDOM NUMBERS.

$0.0 \leq \text{Math.random}() < 1.0$
 ↑ IMPORTANT!

random # $0 \leq \text{guess} \leq \text{maxValue}$;
 integer

$\text{guess} = (\text{int})(\text{Math.random}() * \text{maxValue});$

switch STATEMENT

```

switch (expr)
{
    case a:
        ...
        break;
    case b:
        ...
        break;
    default:
        ...
}

```

fall through behavior

conditional expression

chooses which expression to evaluate based on a boolean

$y = (y < 0.0) ? -y : y;$

$y *= (y < 0.0) ? -1.0 : 1.0;$

? is a VERY LOW PRECEDENCE OPERATOR

$z = |a| + |b|;$

$z = a < 0.0 ? -a : a + b < 0.0 ? -b : b;$

$(a < 0.0) ? (-a) : (((a + b) < 0.0) ? (-b) : b)$

OPERATOR PRECEDENCE & ASSOCIATIVITY

DON'T LET THE COMPILER DO YOUR THINKING!

DEBUGGING