GPSS – General Purpose Simulation System

Discrete Event Simulation Language

Categories of Simulation Software

- General Purpose Languages
  - C, C++, Java
- Simulation Languages
  - GPSS, SIMAN, SLAM, SSF
- Simulation Environments
  - Enterprise Dynamics, Arena, SIMUL8

Features of Simulation Languages

- Some focus on a single type of application
- Built in features include
  - Statistics collection
  - Time management
  - Queue management
  - Event generation

Features of Simulation Environments

- Some focus on one type of application
- Icon based
- Analysis of I/O
- Advanced Statistics
- Optimization
- Support for Experimentation

GPSS – A Transaction Oriented Language

- GPSS – 1961 @ IBM
  - Based on block diagrams
  - Well-suited for queuing models
  - Expensive at first
  - Easily understood GPSS model would require many pages of other language coding to accomplish a similar goal.
  - The language itself collects statistics, produces tabulated results.

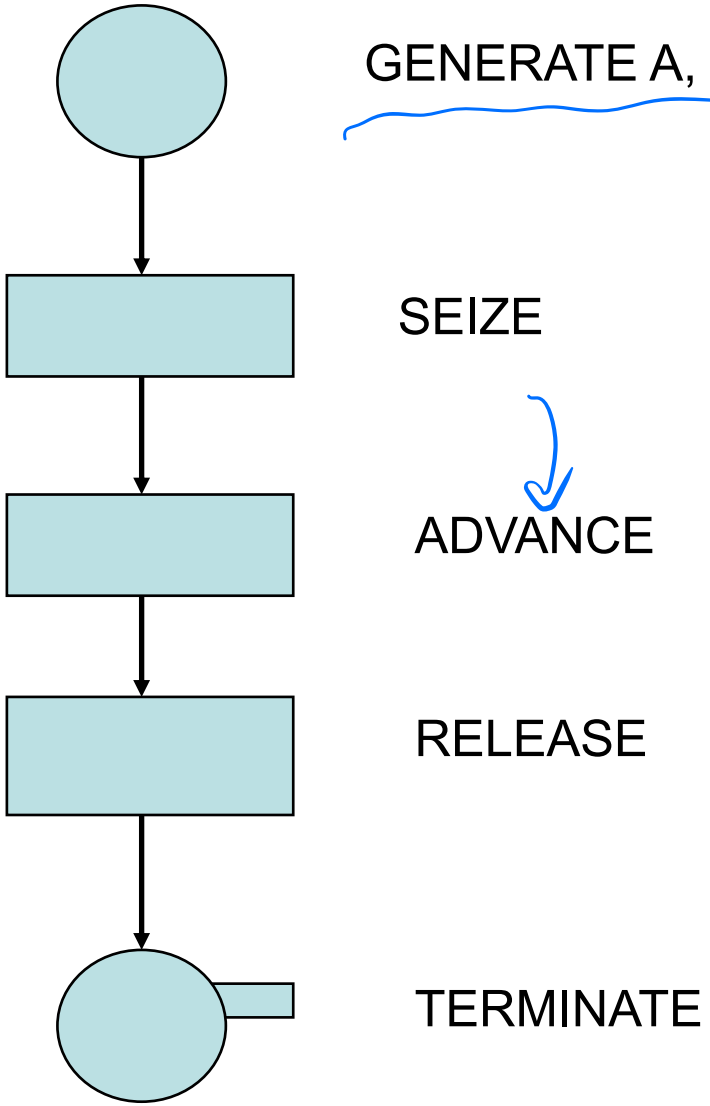
Transactions and Blocks

- Transactions - a set of abstract components of various types.
- Blocks - a set of operators which perform certain actions on the individual components.
- The transaction moves through a sequence of blocks that has been designed to model the system being studied.
- Several different types of equipment components:

  Facility - an entity which is either available for use or is in use by at most one transaction at a time.

Model Building in GPSS

- Block Diagram



Statements

- Blocks
- Definition statements
- Control statements

General format of statements

col 2	col 11	col 22
Label	statement type	parameters (A,B,C,D,E,F,G)

Format for Statements

Columns	Type of information
2-9	Label (if any)
11-20	Operation
22 et seq.	Operands (if any)
> **(last operand's last column, +1)**	Comment (if any)

An Example

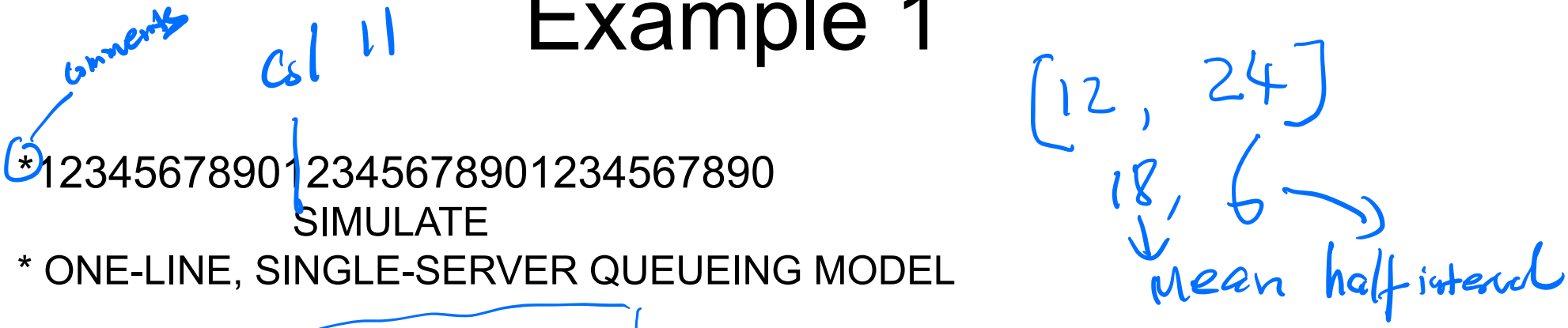
1	Label (2-9)	Operation (11-22)	Operands (22 – 72) →
		GENERATE	15.0, 4.5 Transactions arrive
		TERMINATE	1 Transactions are
*			Destroyed, TC value
*			is reduced by 1

Note: 1. Operands should not extend beyond column **72**.

2. Comments should not extend beyond column **80**.

3. Comments provide documentation and/or leaving blank lines.

Example 1




```
*1234567890123456789012345678901234567890
                    SIMULATE
* ONE-LINE, SINGLE-SERVER QUEUEING MODEL

GENERATE    18,6    ARRIVALS EVERY 18 +- 6 MINUTES
ADVANCE     0.5     HANG UP COAT
SEIZE       JOE     CAPTURE THE BARBER
ADVANCE     15,3    HAIRCUT TAKES 15 +- 3 MINUTES
RELEASE     JOE     FREE THE BARBER
TERMINATE   1       EXIT THE SHOP
*
            START   100
            END
```