



# BridgeLabz

Employability Delivered

Programming  
Constructs –  
Sequence  
Statements

# What is a Programming Construct

A Programming Construct is to control the order/flow in which instructions are executed.

# What is a Programming Statement

In programming languages, the expression which translates to an instruction is called a programming statement or just statement.

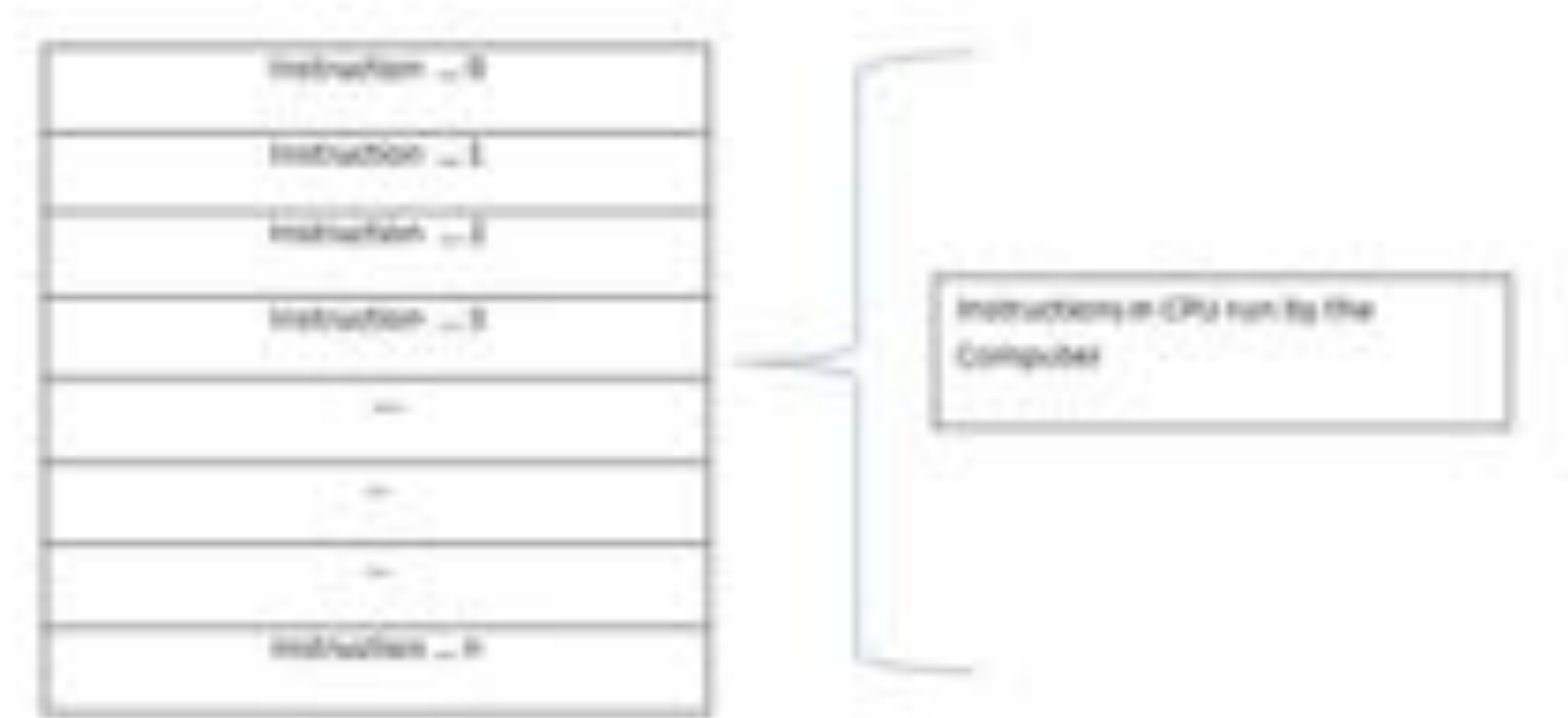
# Programming Constructs Classification

1. Sequences
2. Selection
3. Repetition

# 1. Sequence Statement

A sequence construct tells the CPU (processor) which statement is to be executed next.

# A Program is mainly set of Instructions



# Demonstrating add.sh & Execution Thread

```
GNU nano 2.0.6 File: add.sh

#!/bin/bash -x
x=100;
y=100;
z=$(( $x + $y ))
echo $z
```

add.sh program with following Instructions

1. Variable \$x Declaration and Assignment
2. Variable \$y Declaration and Assignment
3. Addition of \$x and \$y and assigning to variable \$z
4. Displaying \$z
5. NOTE : Arithmetic Expression  *$\$((expression))$*

```
Narayans-MacBook-Pro:TerminalCommands narayan$ nano add.sh
Narayans-MacBook-Pro:TerminalCommands narayan$ ./add.sh
+ x=100
+ y=100
+ z=200
+ echo 200
200
Narayans-MacBook-Pro:TerminalCommands narayan$
```

Terminal showing 4 execution steps indicated by “+”

1. Space allocated in RAM for \$x
2. Space allocated in RAM for \$y
3. CPU fetches value of \$x and \$y from memory. Adds \$x + \$y and stores the value in \$z
4. Display the value of \$z

# Sequences Practice Problems

1. Use Random Function `(( RANDOM ))` to get Single Digit
2. Use Random to get Dice Number between 1 to 6
3. Add two Random Dice Number and Print the Result
4. Write a program that reads 5 Random 2 Digit values , then find their sum and the average
5. Unit Conversion
  - a.  $1\text{ft} = 12\text{ in}$  then  $42\text{ in} = ?\text{ ft}$
  - b. Rectangular Plot of 60 feet x 40 feet in meters
  - c. Calculate area of 25 such plots in acres



# Sequences Practice Problems

- Write a program that takes a date as input and prints the day of the week that date falls on. Your program should take three command-line arguments:  $m$  (month),  $d$  (day), and  $y$  (year). For  $m$  use 1 for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth. Use the following formulas, for the Gregorian calendar (where  $/$  denotes integer division):
  - $y_0 = y - (14 - m) / 12$
  - $x = y_0 + y_0/4 - y_0/100 + y_0/400$
  - $m_0 = m + 12 \times ((14 - m) / 12) - 2$
  - $d_0 = (d + x + 31m_0 / 12) \bmod 7$



**BridgeLabz**

Employability Delivered

Thank  
You