

# Department of Computer Science and Engineering

## **RPA Design and Development (CSE552)-(3-0-0-0)**



**Dr. Bheemappa H**  
**Dept. of CSE**  
**RIT**

# Robotics Process Automation(RPA)

- Robotic process automation (RPA) refers to software that can be easily programmed to do **basic, repetitive tasks** across application.
  - The purpose of RPA is to transfer the process execution from **Humans to Bots**.
  - **Software Bots** can interact with an in-house application, website, user portal to achieve the desire goal.
  - RPA does not require the **development of code**
  - Banking and Finance Process Automation, Data Extraction Process back office process

## Advantages

- Reduced costs.
- High flexibility and improved **ROI (Return of Investment)**.
- Improved data control.
- No requirement for manual data entry.
- minimal chances of error.

# Course Contents

---

## UNIT I

**PROGRAMMING BASICS:** Introduction to Programming, Data and Data Structure, Algorithms, Variables and Arguments, Software Application and Software Development Life Cycle (SDLC), Frameworks and Languages

**AUTOMATION AND RPA:** History of Automation, Automation and its benefits, Introduction to RPA, Automation vs RPA, Process and Flowchart ,RPA Programming Constructs, Robots in RPA, Introduction to Robots, Types of Robots, Benefits and Implementation of RPA

## UNIT II

**RPA TOOL INTRUDUCTION AND BASICS:** RPA Development Life Cycle, How does RPA Work, Challenges in RPA, Variables and Types of Variables, Variables vs. Arguments, Namespaces, and Importing New Namespace.

**CONTROL FLOW ACTIVITY:** Sequences, Control Flow and its types, Decision control-IF, Switch, IF vs Switch, Loops-Do While, While, For each, Other control flow activities - Delay, Break, Assign, Continue and Parallel

# Course Contents

---

## UNIT III

### **DATA MANIPULATION:**

Data Manipulation and Its Importance, String Manipulations, Data Table Manipulations, Collection, Its Types and Manipulations.

**UI AUTOMATION & SELECTORS:** UI interactions, Input actions and Input methods, Containers, Recording & its types, Selectors, Types of Selectors- Full and Partial, Containers and Partial Selectors, Dynamic Selectors

## UNIT IV

**AUTOMATION CONCEPTS AND TECHNIQUES:** Desktop and Web Recording, Extraction and its techniques- Screen scraping, Data scraping and PDF Extraction. Automation Techniques- Workbook and Excel automation (read/write).

**EMAIL AUTOMATION:** Incoming Email automation - Sending Email automation

## Unit V

**ERROR AND EXCEPTION HANDLING:** Errors, Error handling approach, Try Catch, Retry Scope, Exception Handling, Types of Exceptions, Global Exception Handler, Best Practice for Error Handling

# Automation

**Automation** is the technology by which a process or procedure is performed with minimal human assistance.

- Operating equipment such as machinery, processes in factories, boilers.
- Switching on telephone networks, steering and stabilization of ships, aircraft and

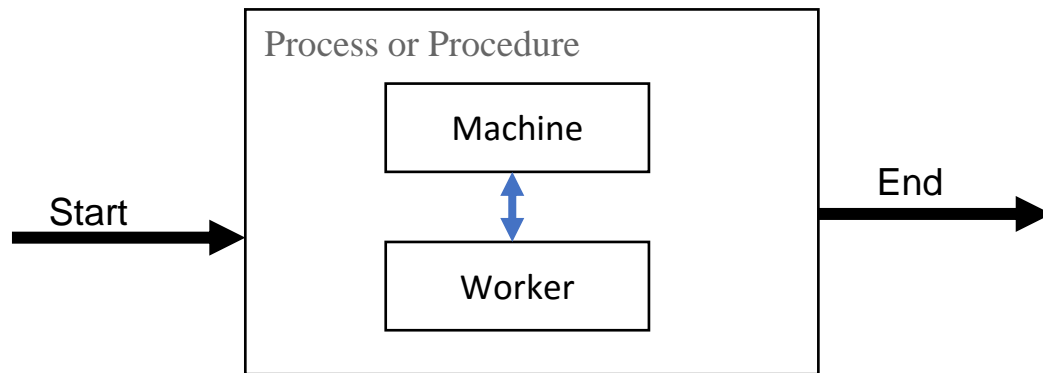


Fig: 1.a General execution of process

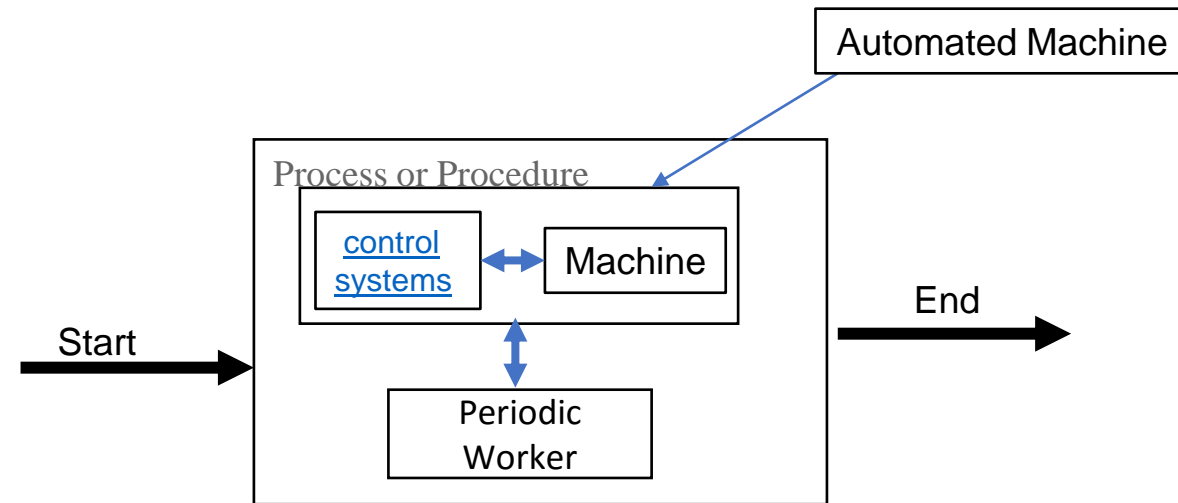
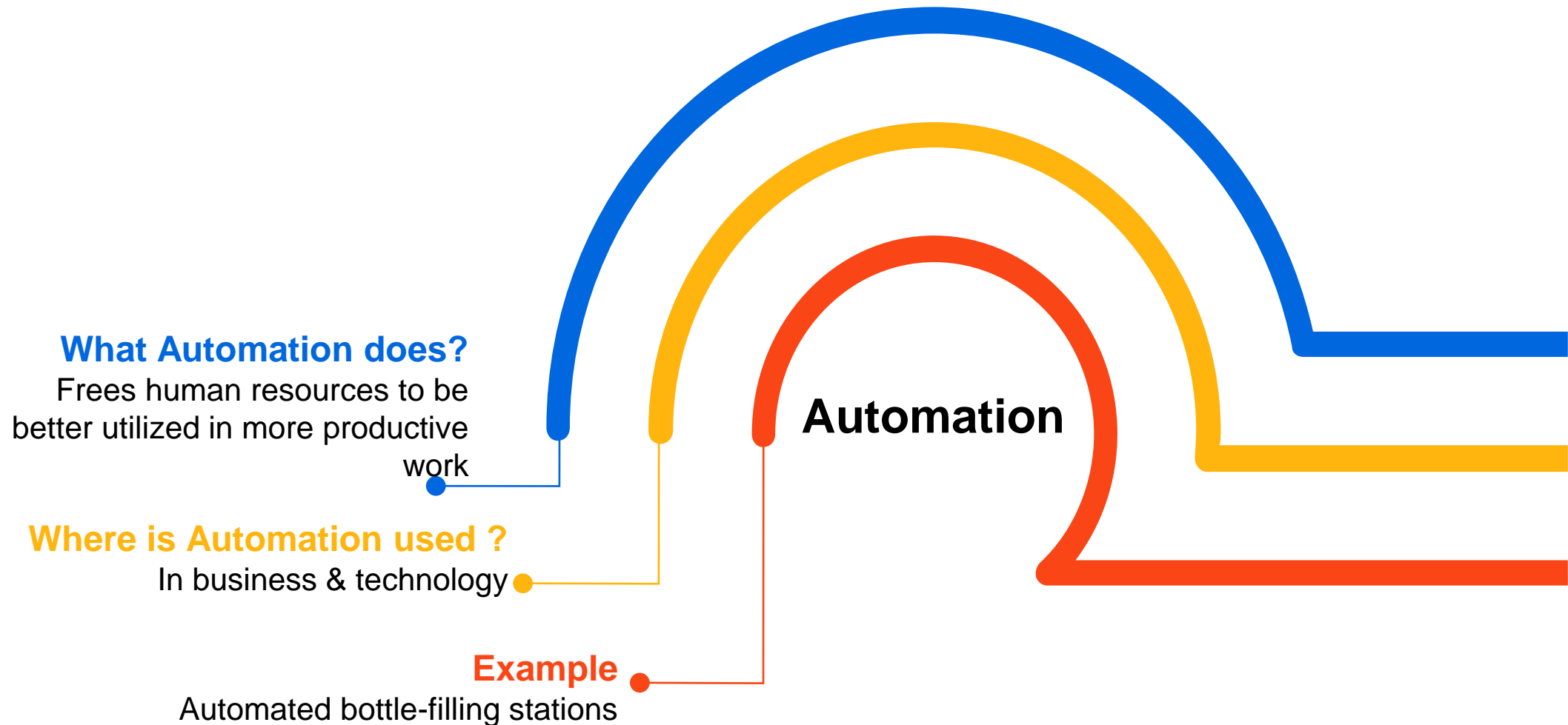


Fig: 1.b Execution of process using Automation

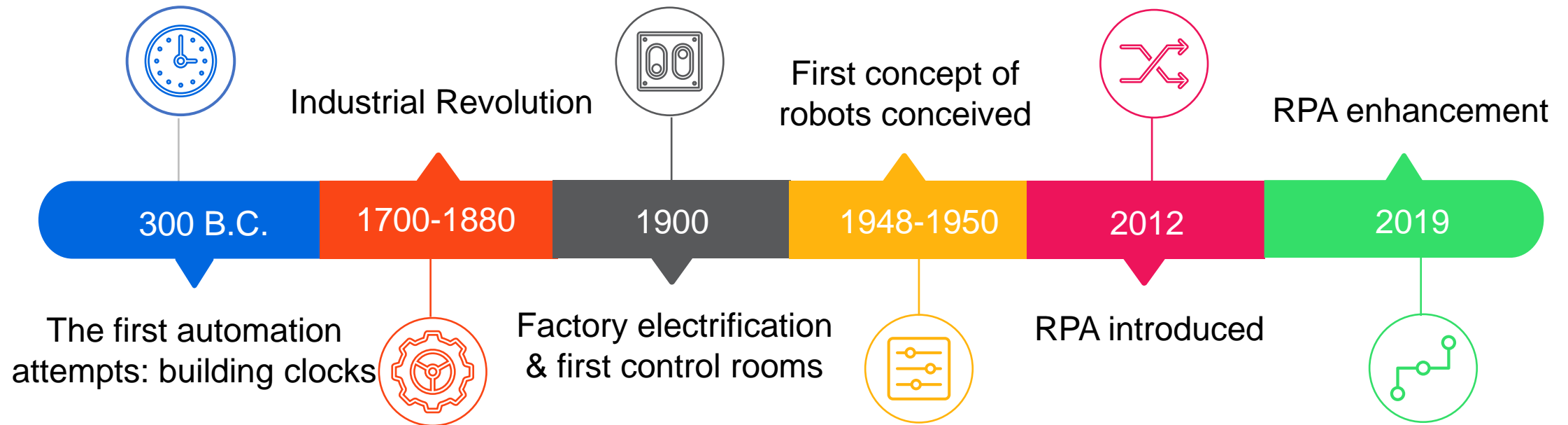
# Introduction to Automation

Automation is the technology by which a process or procedure is executed with minimal human assistance.



# Overview of History of Automation

The history of automation began more than 2300 years ago.



# Introduction to Programming

**Programming** is the process of writing instructions for computer to perform certain tasks.

**Computer program** is sequence of instructions to solve the problem or carry out the task

Characteristics of a good program are:

- Robustness
- Usability
- Portability
- Maintainability

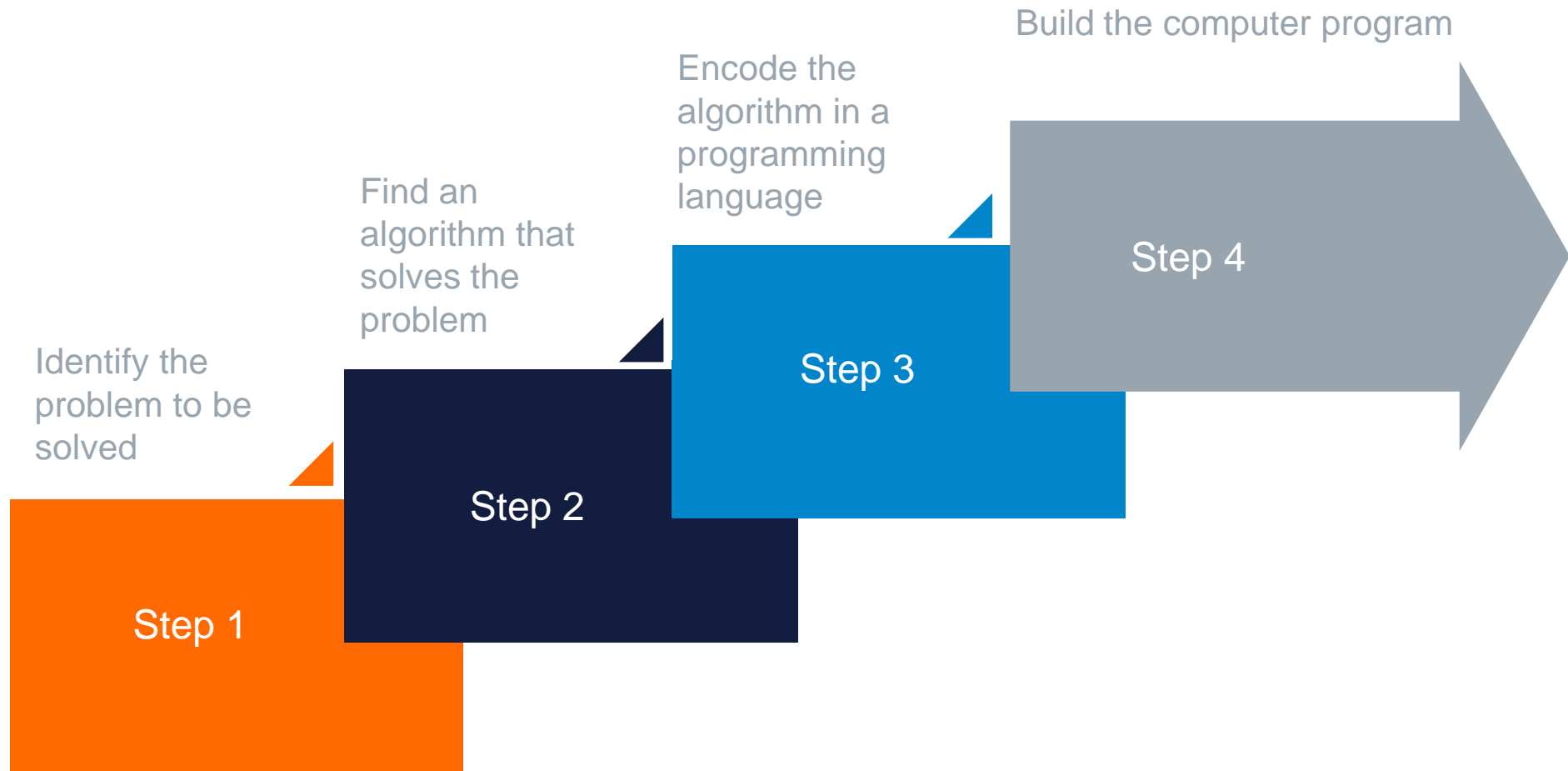


# Methods of Programming

- Procedural
  - Defining set of steps to transform inputs into outputs
  - Translating steps into code
  - Constructed as a set of procedures
  - Each procedure is a set of instructions
- Object-Oriented
  - Defining/utilizing objects to represent real-world entities that work together to solve problem
  - Basic O-O Programming Components
    - Class
    - Object/Instance
    - Properties
    - Methods

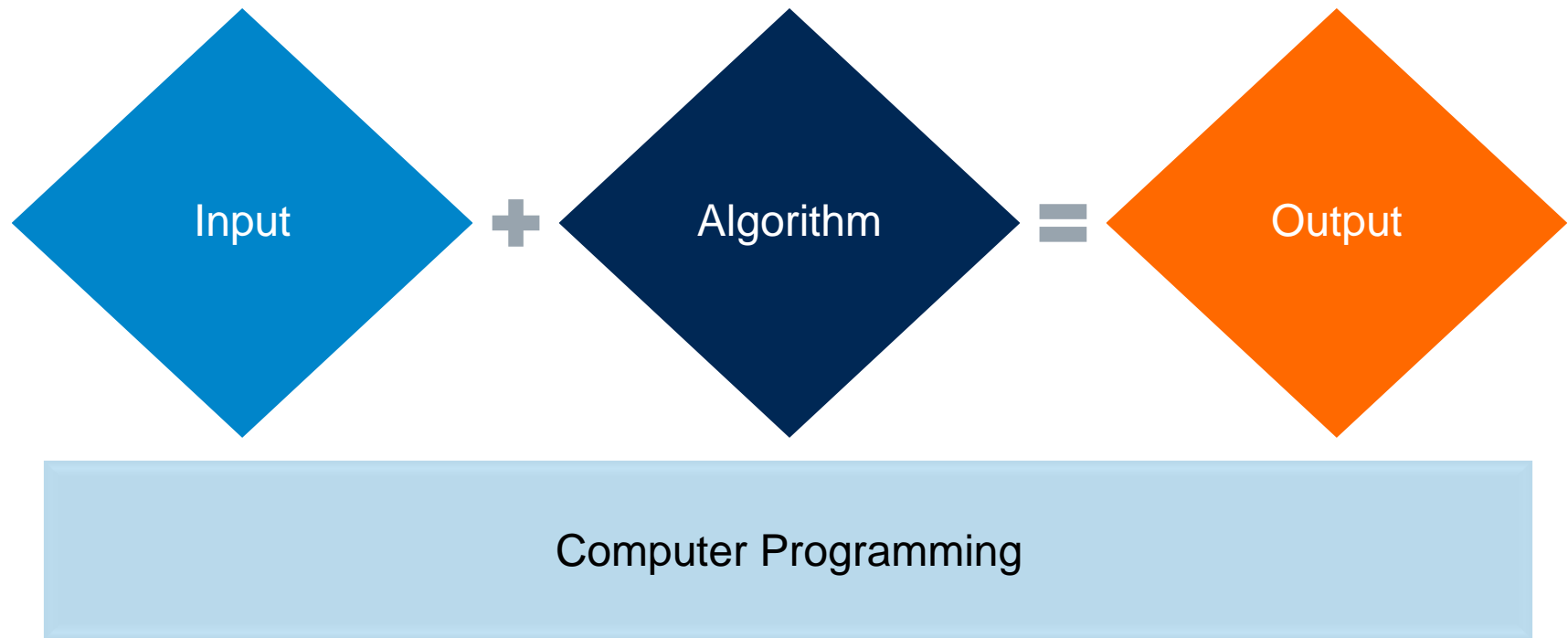
# Steps for Creating a Computer Program

Building blocks of a computer program.



# Key Components of Programming

Input, output and algorithm form an integral part of computer programming.



# Data

Data is a piece of digital information. Data is a set of variables/values that can be processed by any computing machine.

## **Examples**

Weights, Number of items sold, Employee names

## **Information**

When the raw data is processed it turns into information.

.

## **Storage Form**

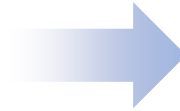
At machine level, all data is stored in the form of 0 and 1.

# Data Storage

There are two ways of storing data:

## Temporary

- Stored in computer's temporary memory



## Volatile memory,

- Example : RAM, or **memory**
- Data **storage** device does not retain its data when the device is unpowered
- operates faster than persistent storage

## Permanent

- Stored permanently for current and later use
- Stored in files



## Non-Volatile memory,

**Persistent storage**, is any computer data **storage** device that retains its data when the device is unpowered

SSD/HDD  
Memory card

# Types of Data

On the basis of structure and uniformity, data can be categorized as:

**Structured Data:** Its highly organized and easily understood by machine language.

**Unstructured Data:** it cannot be processed and analyzed using conventional tools and methods.

Structured Data



**For example:** Spreadsheet, Online forms,  
airline reservation systems, inventory control,  
sales transactions, and ATM activity.

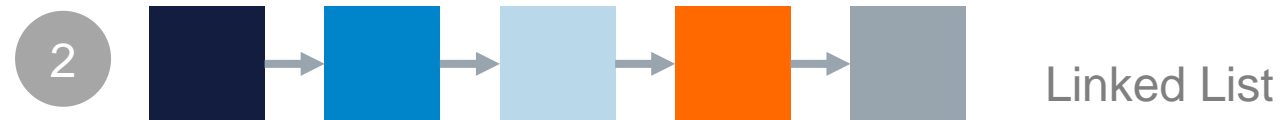
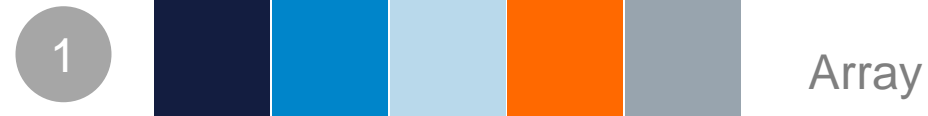
Unstructured Data



For example: Email,  
Data from Facebook,  
Twitter, LinkedIn.

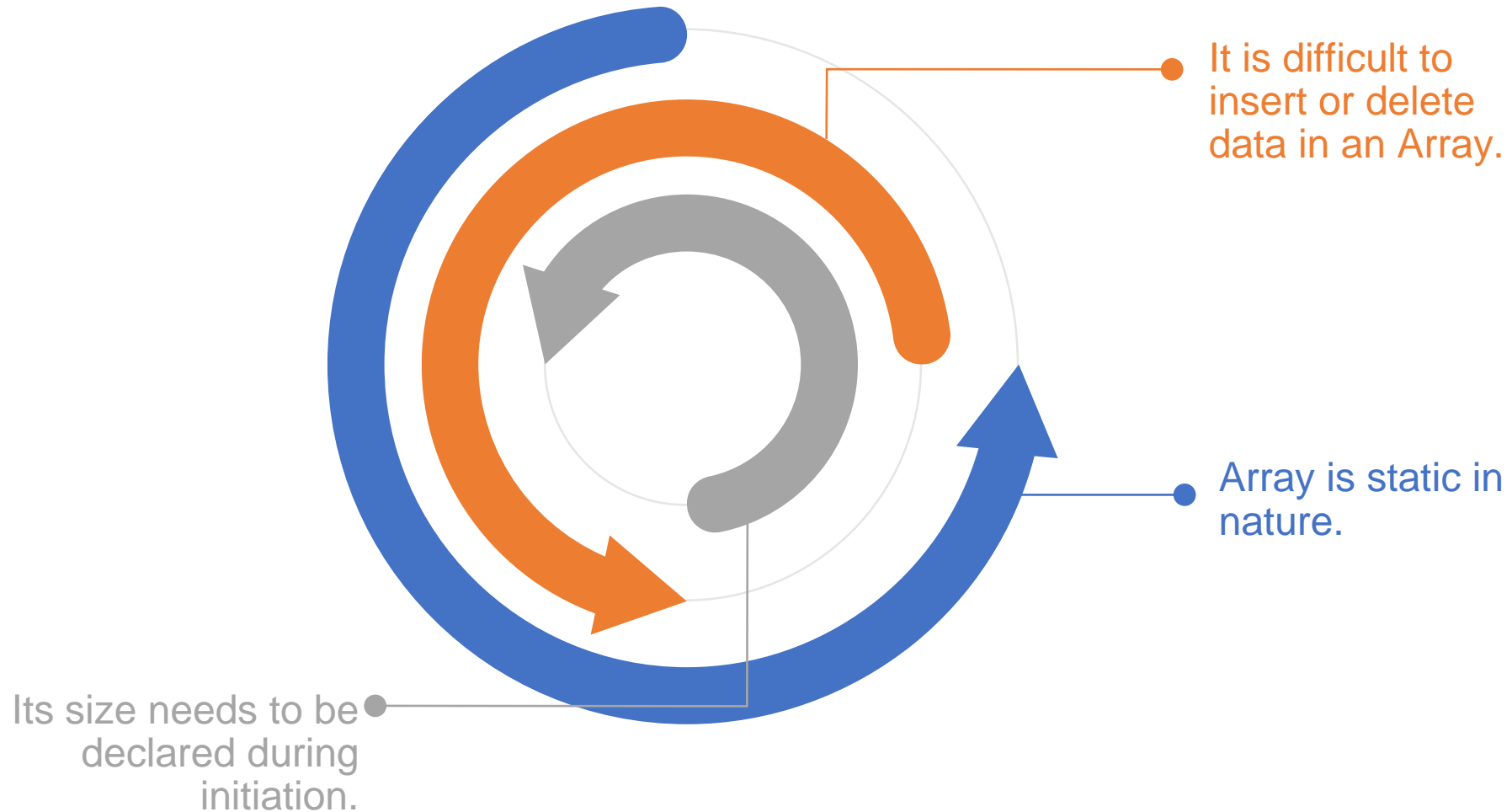
# Types of Data Structures

A **data structure** is a particular way of organizing data. There are five different types of data structures:



# Array

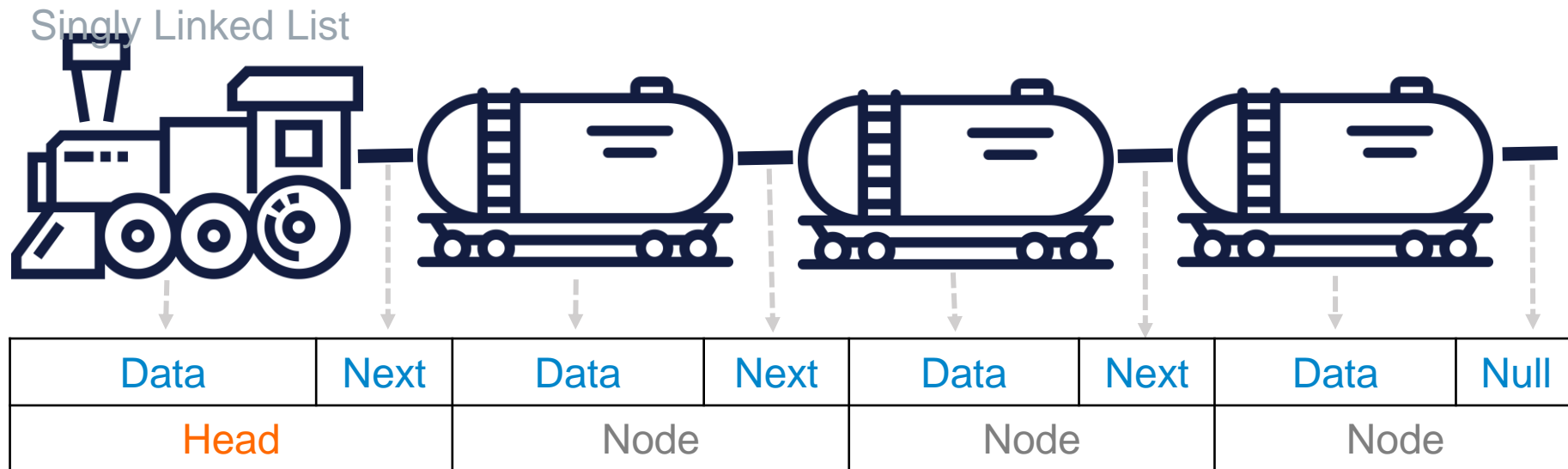
**Array** is a data structure which stores homogeneous elements at logical locations.





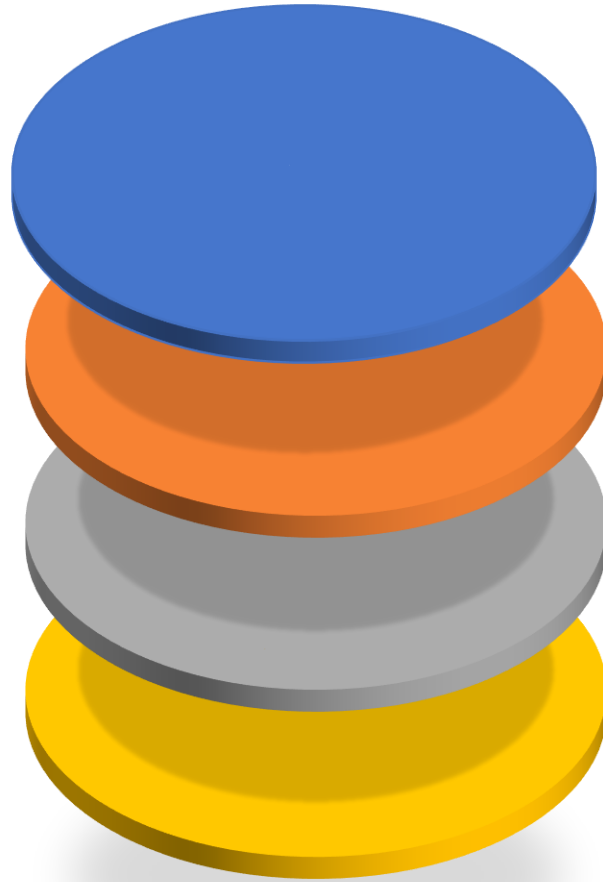
# Linked List

A **Linked List** is a linear data structure containing a set of records linked together by links.



# Stack

Consider an example of books stacked over one another in the library.

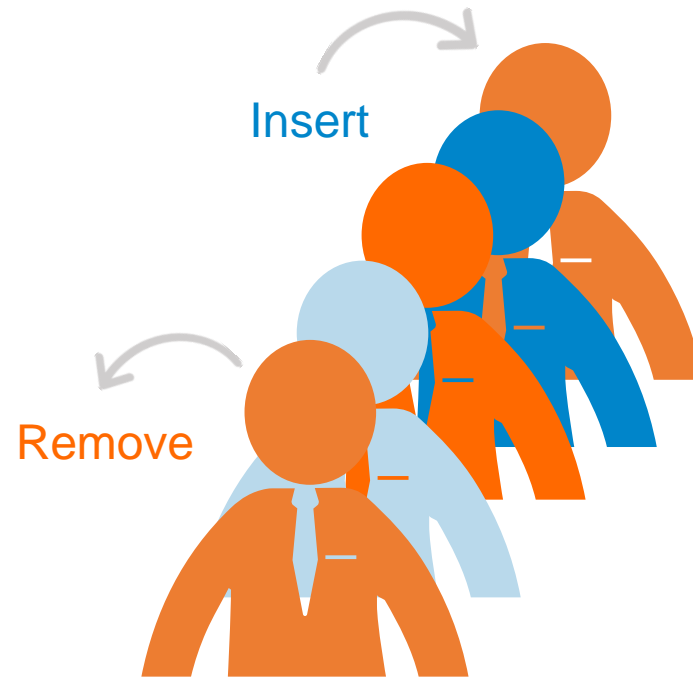


A **stack** is a linear data structure which follows a particular order in which the operations are performed.

This order may be LIFO (Last In First Out) or FILO (First In Last Out).

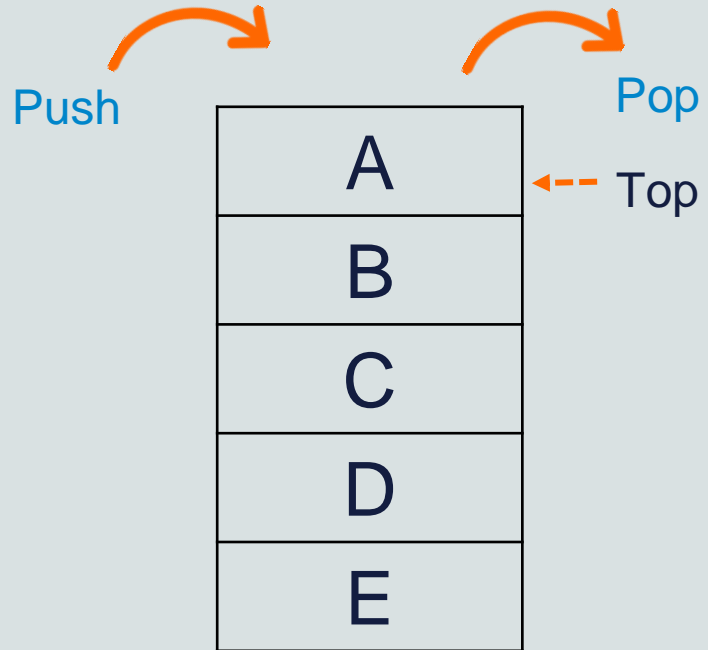
# Queue

A **queue** is a linear data structure in which the operations are performed in First In First Out (FIFO) order.

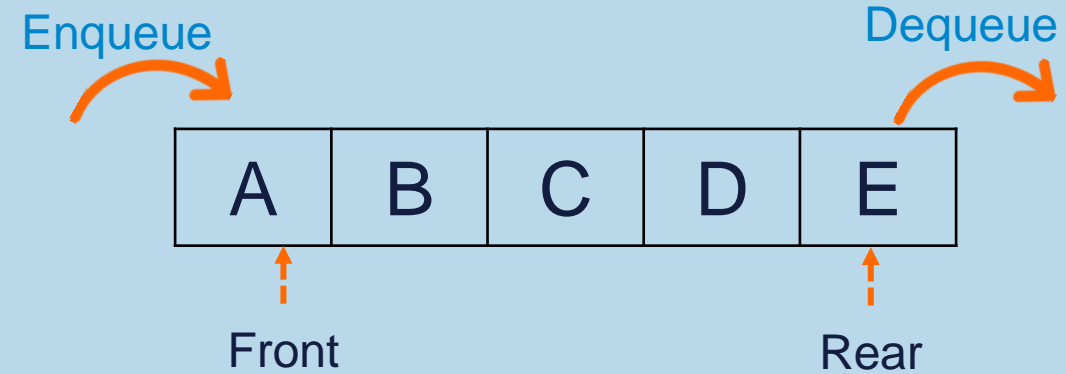


Can you give a real life example of a Queue structure?

# Stack vs. Queue



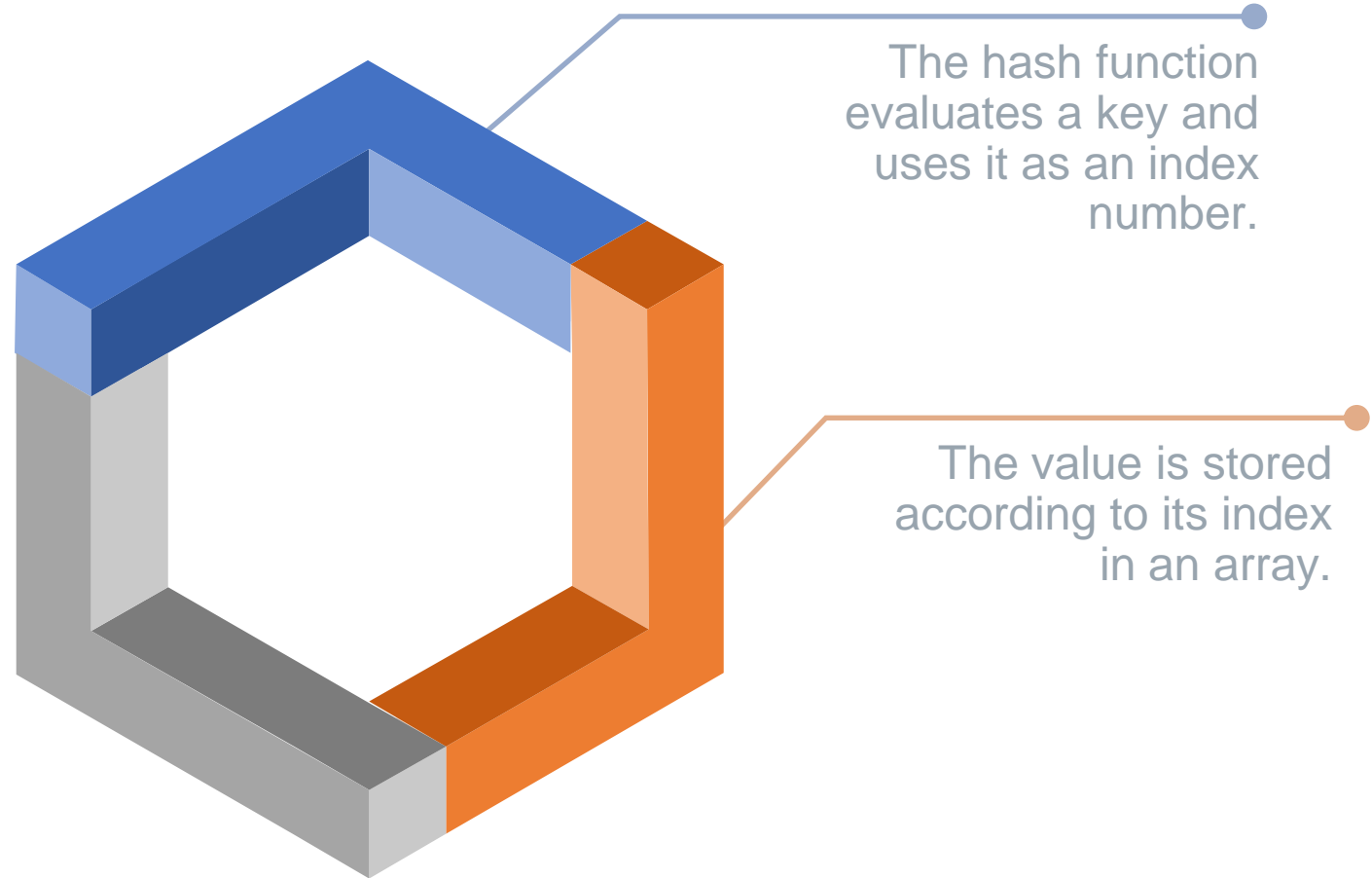
Stack



Queue

# Dictionary

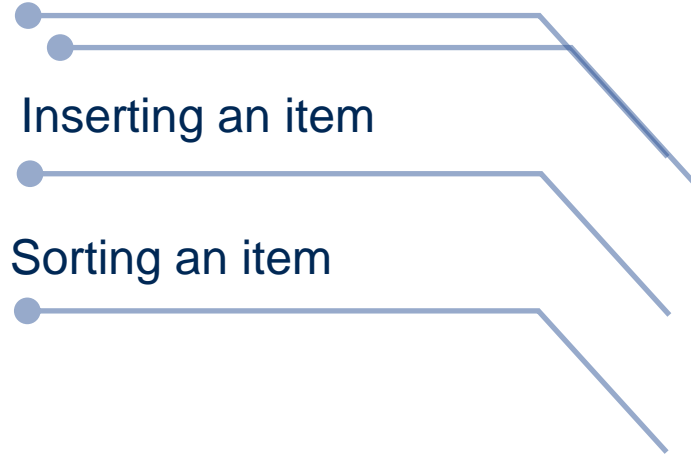
A **dictionary** consists of a key and a value.



# Algorithm

Finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems

Searching for an item



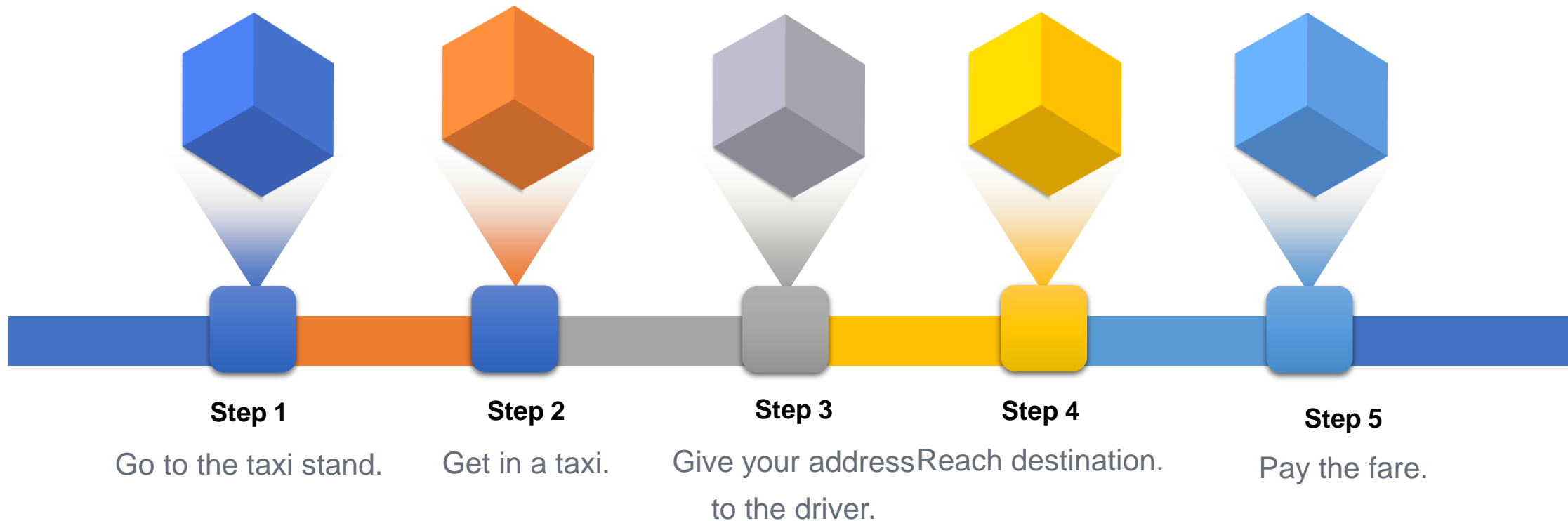
Inserting an item

Sorting an item

Algorithms are used to link data structures together to create a program that solves a certain task. They are also used to manipulate data in various ways, such as:

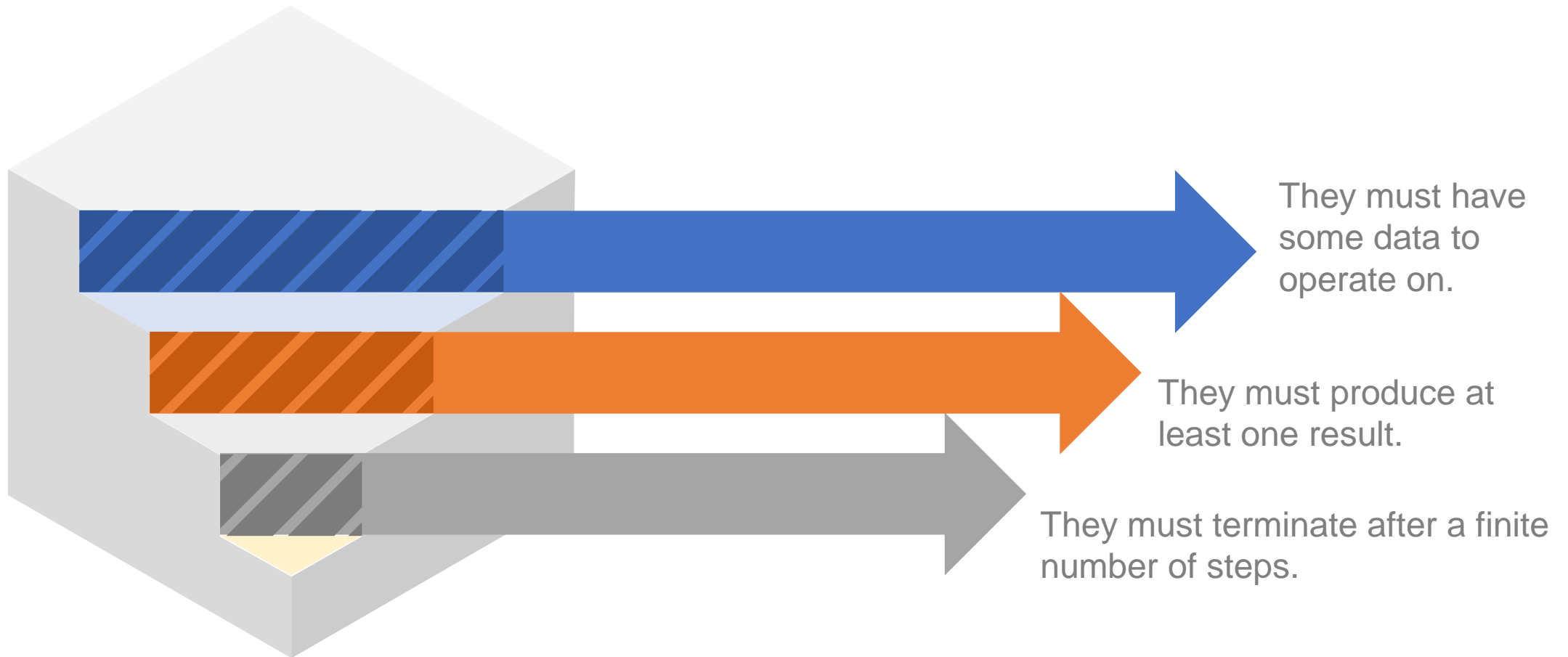
# Algorithms in Day-to-Day Life

Algorithms can be associated with things that we do in our daily life. An example of an algorithm for a taxi ride is as follows:



# Algorithms in Programming

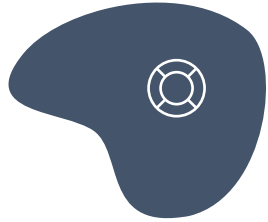
Algorithms have certain features:





# Types of Algorithms

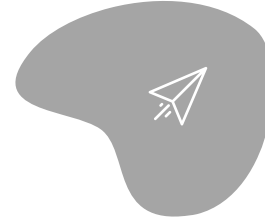
Algorithms are divided into following generic categories:



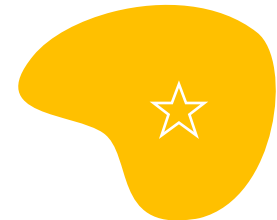
Searching



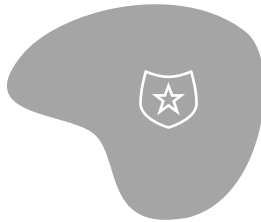
Sorting



Compression



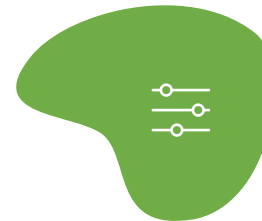
Parsing



Encoding




Geometric



Pattern  
Matching

# Variables


**Variables** are storage containers for data that can be used later throughout the program.



**Counter**  
Keep track of a certain repetitive action, like clicking on an item



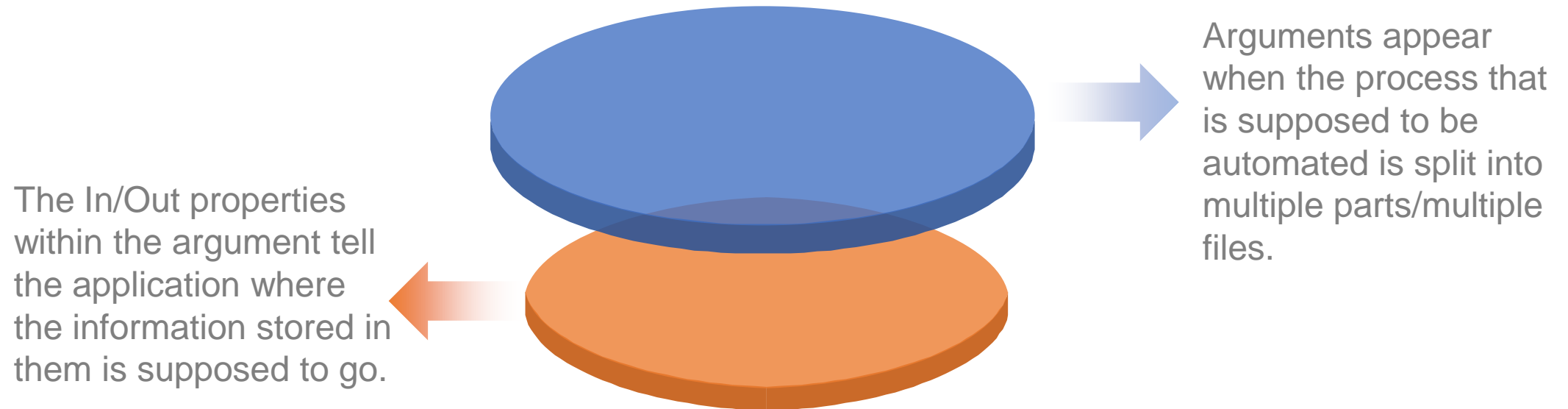
**Comments**  
Store the comments for each action item



**Organized Items**  
Store organized items of interest to the users that clicked on them

# Arguments

An **argument** is a variable that can store a value which can be used later to pass values between files.



# Uses of an **Argument**

**Arguments** are used in breaking up large bits of code into smaller re-usable bits of code.

- **Save Time**

Codes can be re-used

- **Pass Values**

Can be passed between the projects and programs

- **Use in RPA**

Can store and pass dynamic data to variables

# PROGRAMMING BASICS:

- Introduction to Automation
- **Introduction to Programming,**
- **Data and Data Structure,**
- **Algorithms**
- **Variables and Arguments**
- **Software Application**
- **Development Life Cycle (SDLC),**
- **Frameworks and Languages**

# Sequence

A **sequence** is composed of a series of statements which are executed one by one from top to bottom.

# Control Flow in Program

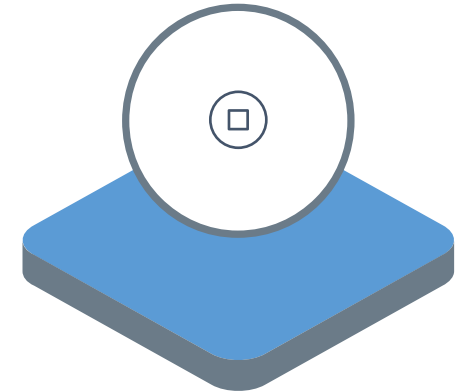
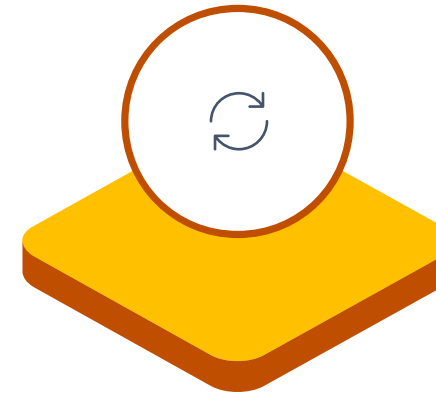
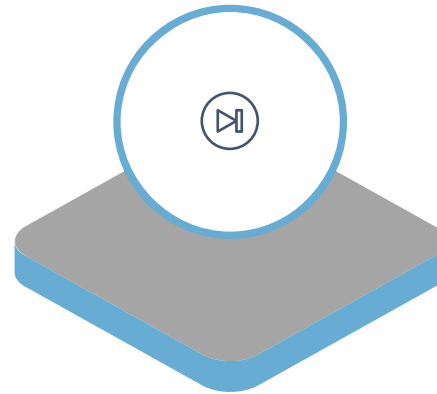
**Control flow** is the order in which individual statements, instructions or steps are executed or evaluated in a software program. Flow in a program can be controlled by:

Executing **only if** some condition is met

Executing and **returning**

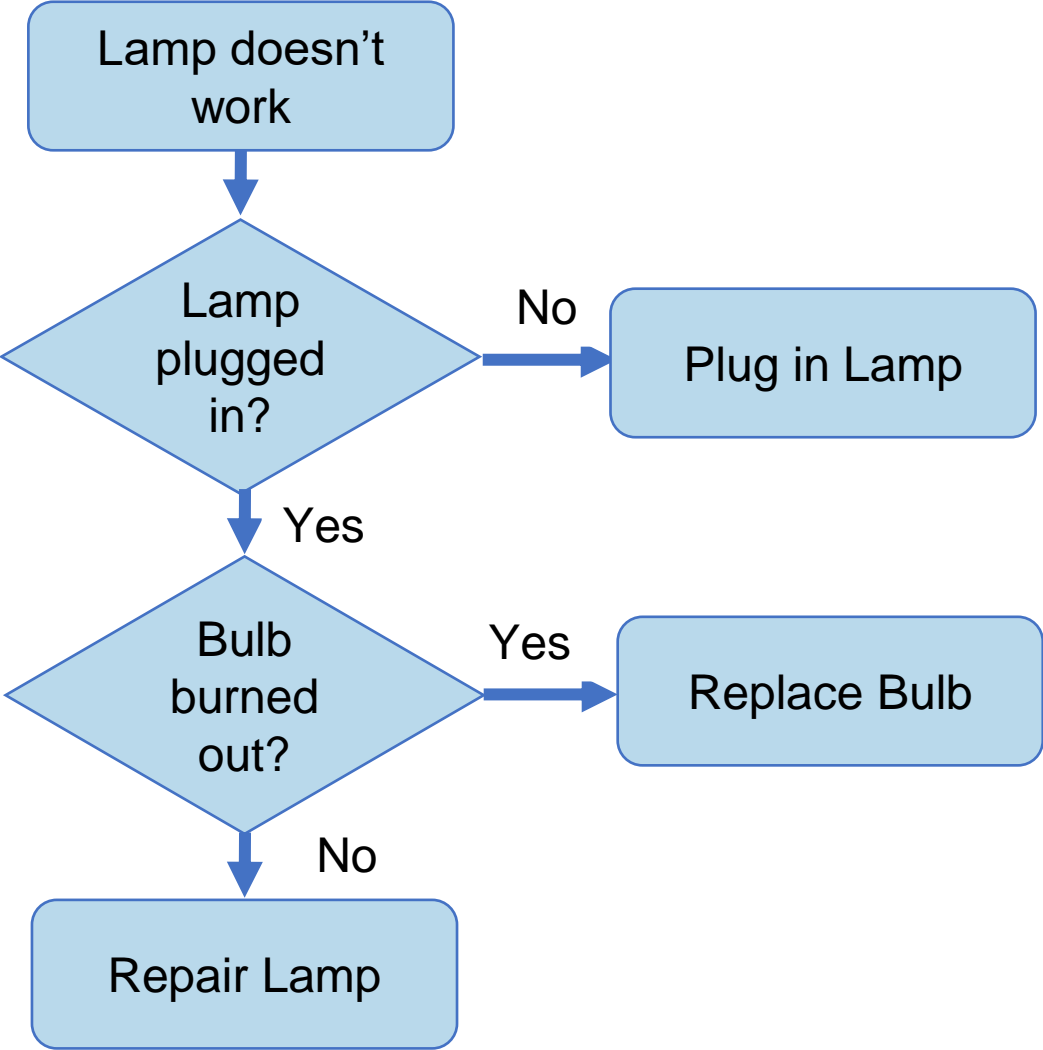
Executing **until** some condition is met ✓



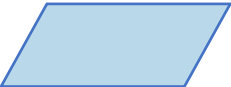

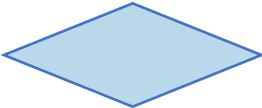
**Preventing** any further execution



# Flowchart

A **flowchart** depicts the flow of a program.



Symbol	Name
	Start/end
	Arrows
	Input/output
	Process
	Decision



# Software Application

An **application** is a computer software package that performs a specific function directly for an end-user or for another application.



Application Software		
Word Processing	Spreadsheets	Presentations/ Web Browsers
Gaming	E-mail	Databases
System Software		
File Management Tools	Operating System	Utilities

# Software Development Life Cycle (SDLC)

The following figure is a graphical representation of the various stages of a typical [SDLC](#).

**SDLC** : Its process used by the software industry to design, develop and test high quality software's.

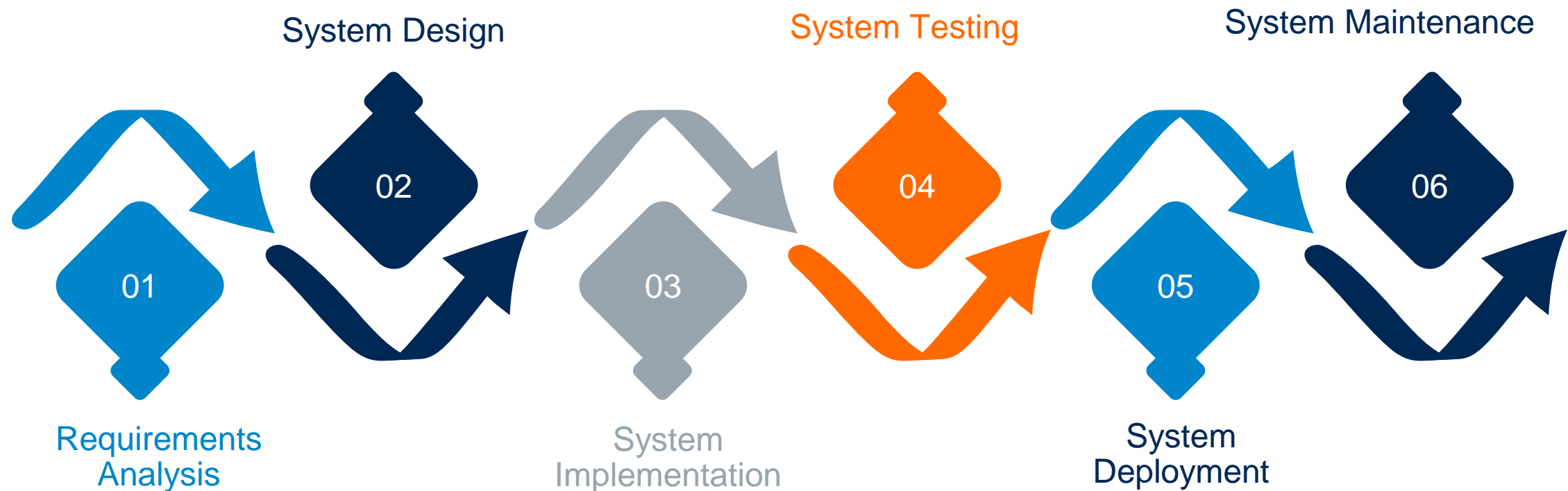


# SDLC Models

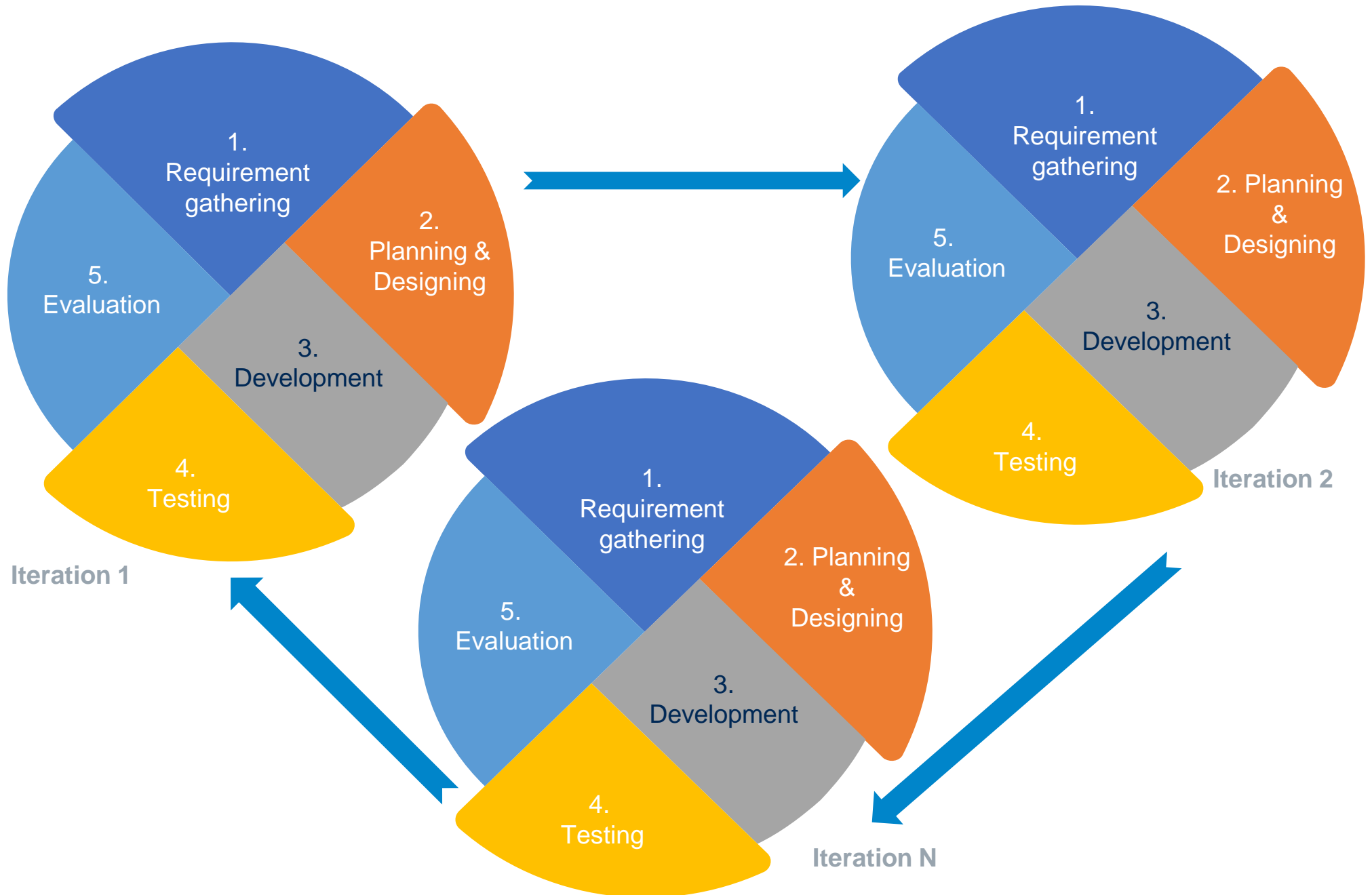
# Waterfall Model

Waterfall method focuses on gathering all of the customer requirements at the beginning of the project so that every step of the SDLC can be planned.

-Highly sequential and can be broken down into seven distinct phases



# Agile Model



# Agile Model

The Agile model provides continuous iteration of development and testing in the software development process

- Development and testing activities are concurrent(unlike in Waterfall)
- Rapid Application Development

It guarantees that the quality of the development is clearly maintained.

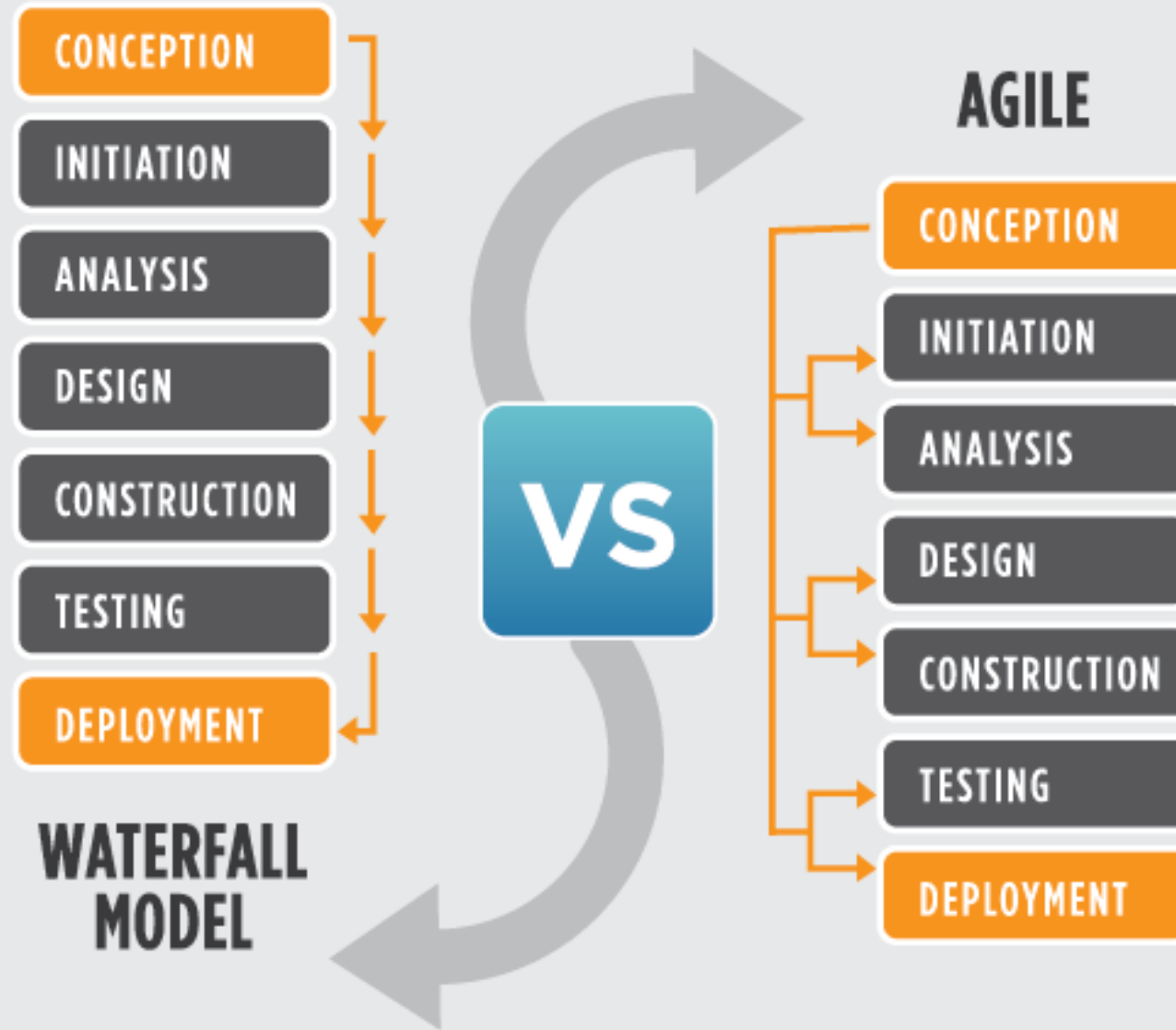
## **Agile disadvantages**

Agile is not useful for small development projects.

Agile meetings require the presence of an expert to take important decisions.

Agile projects can easily go off track if project managers are not sure what outcome they want.

The cost of implementing Agile is little more compared to other methodologies



ILLUSTRATED BY SEGUE TECHNOLOGIES

