# Rust College: Coding Influencer







Pn

### Table of Contents



THE SWIFT PROGRAMMING LANGUAGE PAGE 1 | DATA TYPES PAGE 2 | BASIC OPERATORS PAGE 3 | VARIABLES AND CONSTANTS PAGE 4 | COLLECTION TYPES PAGE 5 | CONTROL FLOW PAGE 6 | FUNCTIONS PAGE 8 | METHODS PAGE 9 | TYPE CASTING PAGE 10 | ERROR HANDLING PAGE 11 | STRUCTURES AND CLASSES PAGE 12 | INHERITANCE PAGE 13 | COMMENTS PAGE 14 | REFERENCES PAGE 15

Pn

### The Swift Programming Language

#### WHAT IS SWIFT?

"Swift is a new programming language for iOS, macOS, watchOS, and tvOS app development. Nonetheless, many parts of Swift will be familiar from your experience of developing in C and Objective-C [1]."

--Swift.org

### Data Types

#### WHAT ARE DATA TYPES?

Data types are classifications that specifies which type of value a variable has and what type of mathematical, relational, or logical operations can be applied to it without causing an error [2].

Basic Data Types [1]:

- Int for integers
- Double and Float for floating-point values
- Bool for Boolean values
- String for textual data

### **Basic Operators**

#### WHAT ARE BASIC OPERATORS?

Operators are a special symbols or phrases that you use to check, change, or combine values [1].

Internet C perator(b)	
assignment operator (=)	let a = 20
	let $(x, y) = 4, 5$
addition operator (+)	let $j = 1 + 5$
	<pre>let m = "hello, <u>" +</u> "world" // "hello, world"</pre>
subtraction operator (-)	let $j = 1 - 5$
multiplication operator (*)	let j = 1 * 5
division operator (/)	let $j = 10 / 5$
remainder operator (&)	let $j = 9 \% 4$
unary minus operator	let two = 2
	let minusTwo = -two
	let plusTwo = -minusTwo
unary plus operator	let minusFive = -5

Comparison Operator(s)	Example
equal to (==)	m == n
not equal to (!=)	<u>m !</u> = n
greater than (>)	m > n
less than (<)	m < n
greater than or equal to (>=)	m >= n
less than or equal to (<=)	m <= n

### Variables and Constants

#### WHAT ARE VARIABLES AND CONSTANTS?

Variables are elements, features, or factors that is liable to vary or change. Constants are data values that stay the same every time a program is executed [1].

let maximumNumberOfTimesToPlayGame = 10 //constant
var currentNumberOfAttempts = 0 //variable

## **Collection Types**

#### WHAT ARE COLLECTION TYPES?

There are three collection types for storing collections of values. They are as follows: arrays, sets, and dictionaries. Arrays are ordered collections of values. Sets are unordered collections of unique values. Dictionaries are unordered collections of key-value associations [1].

var toyList: [String] = ["blocks", "video games", "toy cars"] // array
var toyList: Set<String> = ["blocks", "video games", "toy cars"] // set
var toyList: [Int: String] = [1: "blocks", 2: "video games", 3: "toy cars"]
// dictionary

#### WHAT IS A CONTROL FLOW?

Control flow statements order the execution of function calls, instructions, and statements are executed or evaluated when a program is running. The following are control flow statements: *while*, *repeat-while*, *if*, *guard*, *for-in*, *and switch* [1].

```
for-in loop
    let toys = ["blocks", "video games", "toy cars"]
    for toyin toys {
        print("I like \(toy)!")
    }
while loop
    while condition {
        statements
```

}

## **Control Flow**

```
repeat-while loop
repeat {
statements
} while condition
```

```
if statement
    var numberOfSteps = 30
    if numberOfSteps <= 32 {
        print("You have exceeded the limit.")
    }</pre>
```

```
switch statement
switch some value to consider {
    case value 1:
        respond to value 1
    case value 2,
        value 3:
        respond to value 2 or 3
    default:
        otherwise, do something else
}
```

### Functions

#### WHAT ARE FUNCTIONS?

*Functions* are self-contained chunks of code that perform a specific task [1].

```
func greet(world: String) -> String {
    let greeting = "Hello, " + world + "!"
    return greeting
}
```

### Methods

#### WHAT ARE METHODS?

*Methods* are functions that are associated with a particular type [1].

- *Instance methods* are functions that belong to an instance of a particular class, structure, or enumeration [1].
- *Type methods* are methods you define methods that are called on the type (i.e., data type) itself [1].

## Type Casting

#### WHAT IS TYPE CASTING OR "CASTING"?

*Type casting* is a way to check the type of an instance variable, or to treat that instance variable as a different superclass or subclass from somewhere else in its own class hierarchy [1].

## **Error Handling**

#### WHAT IS ERROR HANDLING?

*Error handling* is the process of responding to and recovering from error conditions (i.e., incorrect input values, etc.) in your program [1].

•

### Structures and Classes

#### WHAT ARE STRUCTURES AND CLASSES?

*Structures* and *classes* are general-purpose, flexible constructs that become the building blocks for objects. They have defined properties and methods to add functionality using the same syntax you use to define constants, variables, and functions [1].

#### WHAT IS INHERITANCE?

A class can *inherit* (i.e., be an extended instance of class) methods, properties, and other characteristics from another class [1]. Thus, you can different types of living spaces, bicycles, dogs, cats, etc.

### Comments

#### WHAT ARE COMMENTS?

Comments are non-executable notes added to code to document functionality [1]. There are two types of comments as follows:

single-line comments (//): // This is a comment.

multiline comments (\*/) /\* This is also a comment but is written over multiple lines. \*/

# References

[1] Apple Inc., "Swift.org," 2022. [Online]. Available: https://www.swift.org/website/. [Accessed 2 June 2022].



# Rust College Center for Cybersecurity

The Rust College Center for Cybersecurity provides a computer science program with a concentration in cybersecurity (i.e., defense and operations). Students and faculty are provided opportunities to collaborate on interdisciplinary research and perform outreach with other institutions.







# Rust College Cybersecurity Influencer









FIRST PRINCIPLES PAGE 1 | DOMAIN SEPARATION PAGE 2 | PROCESS ISOLATION PAGE 3 | RESOURCE ENCAPSULATION PAGE 4 | LEAST PRIVILEGE PAGE 5 | LAYERING PAGE 6 | ABSTRACTION PAGE 8 | INFORMATION/DATA HIDING PAGE 9 | MODULARITY PAGE 10 | SIMPLICITY PAGE 11 | MINIMIZATION PAGE 12

## First Principles

# WHAT ARE THE FIRST PRINCIPLES?

Building a secure system is a design problem, meaning no de-facto recipe is to do so. However, in the absence of methodical techniques, experience has contributed to a set of first principles. The principles are fundamental, foundational propositions regarding what qualities of a system contribute to cybersecurity. Furthermore, these principles guide tradeoffs during system design that contribute to security [2].

### Domain Separation

### DEFINITION

- In a computer, *domain* refers to a collection of data or instructions that warrant protection. Outside of a computer, a domain refers to an area of responsibility or control [2].
- Separating domains allows for enforcement of rules governing the entry and use of domains by entities outside the domain [2]
- During system testing, test data should be separated from "real" data, such as personal information. Such separation avoids unauthorized or accidental disclosure of personal or sensitive data [2].

### EXAMPLE(S)

Most computer processors run in two states. The supervisor domain and the user domain. When in the supervisor domain, the processor can directly access memory (i.e., RAM) or manipulate access control tables in a primitive file system. When in the user domain, the processor cannot access memory that belongs to other programs or the operating system.

A virtual machine (or a container) is a domain that is separate from other virtual machines (or containers) [2]

### **Process Isolation**

### DEFINITION

Ensuring that programs or operating systems run entirely separate from other programs or operating systems to control system resource memory access.

Process isolation is frequently used in web browsers to separate multiple tabs and protect the core browser itself should a process fail. [2, 3]

- A word processor, a database, and a browser running on a computer all run in different address spaces. Process isolation ensures that each one cannot influence the other address space [2].
- A non-technical example of process isolation is when a prosecutor and defense attorney run their cases in court. It would be a problem if either had access to each other's work. Keeping their work separate protects it from misuse by the other party [2].

### Resource Encapsulation

### DEFINITION

The process of separating an entity (system, object, or hardware) to include and isolate its data.

 In object-oriented programming, encapsulation is the inclusion within a program object of all the resources needed to function – basically, the methods and the data [2, 3].

- A word processor, a database, and a browser running on a computer all run in different address spaces. Process isolation ensures that each one cannot influence the other address space [2].
- A non-technical example of process isolation is when a prosecutor and defense attorney run their cases in court. It would be a problem if either had access to each other's work. Keeping their work separate protects it from misuse by the other party [2].

## Least Privilege

#### DEFINITION

The principle of allowing entities (people, processes, devices) only the capabilities necessary to accomplish their assigned duties and functions.

 The term need-to-know is a restrictive information policy often used in the military. Meaning you share information only with the individuals who need to know, only the facts they need to know when they need to know them, and nothing more [2, 3].

- The principle of allowing entities (people, processes, devices) only the capabilities necessary to accomplish their assigned duties and functions.
- The term need-to-know is a restrictive information policy often used in the military. Meaning you share information only with the individuals who need to know, only the facts they need to know when they need to know them, and nothing more [2, 3].

## Layering

### DEFINITION

The process of providing multiple layers of protection or controls between critical data and attackers; layered security can be considered one step of a defense-indepth strategy.

- •A security solution to protect your home computer may include – an antivirus, a firewall, parental controls, and privacy controls.
- In computer programming, layering is the programming organization into separate functional components that interact in some sequential and hierarchical way. Each layer usually has an interface only to the layer above and below it [2, 3].

- A moat is an outer layer that protects a castle. The next layer that an intruder has to go through is the high walls. All of this has to be done by the intruder while avoiding the watchful guards. Finally, the intruder needs to scale the inner walls before getting to the king's treasure [2].
- Firewall, intrusion detection systems, internal encryption, access control, and personnel controls are examples of layers typically employed to protect enterprise data and programs [2]

## Abstraction

### DEFINITION

The principle that the interface of a hardware or software component should be independent of its implementation.

• In Object Orientated Programming, objects are used to represent complex data structures [2, 3].

- The gauges in a car are an abstraction of the car's performance [2].
- A map is an abstraction of an area [2].
- A model airplane is an abstraction of a real airplane and may be used to test aerodynamics [2].



## Information/Data Hiding

### DEFINITION

The principle of keeping information inaccessible except within the process itself.

- The programming concept of making data private rather than public.
- No one can view students' grades except the teacher and the students [2, 3].

- A stack data structure exposes the data only at the top of the stack using simple push and pop instructions. The operating system applies access control to different regions of the stack [2].
- Websites don't need to load all of a user's data to show a list of usernames - they only need the username, the rest of the record fields can be hidden [2].

# Modularity

### DEFINITION

The process of separating functionality into independent pieces to ensure each piece performs a separate function and keeps its data.

- Using functions or methods in programming is an example of modularity.
- Modularity within the system architecture enforces security by keeping operating system functions separate and unique.
- The modular design focuses on building small, carefully crafted components throughout the application [2, 3].

- Electronic circuits [2].
- Lego blocks [2].
- Network nodes [2].

# Simplicity

### DEFINITION

The quality of designing programs, systems, and processes to be free of complexity, easier to test, easier to operate, easier to protect.

• A more straightforward system design will reduce the attack surface area and make it easier to secure the system [2, 3].

- Interface designs that allow correct application of security features [2].
- Intuitive and straightforward access control rules [2]
- Easy to follow and maintain program statements [2].
- Network nodes [2].

## Minimization

### DEFINITION

Keeping all design and functionality aspects to a minimum, reducing needless size and complexity.

- Data minimization is the practice of limiting the collection of information to only that which is directly relevant and necessary to accomplish a task. This policy will also reduce exposure in the event of a breach.
- System minimization implies the practice of only running software, applications, or services necessary to perform the required function. This strategy not only increases security but also can improve performance and save storage space.

- Turn off unnecessary features [2].
- Block unnecessary ports using a firewall [2].
- Reduce the amount of code [2].

# References

 Anonymous "INSPIRING THE NEXT GENERATION OF CYBER STARS,", vol.
 2021, no. June 8,.
 M. Hale, "Introduction to Cybersecurity First Principles,", vol. 2021, no. June 8,.
 Anonymous "2020 GenCyber Grant Call for Proposal,", October 02,.
 Anonymous "Carmen Sandiego (2019 Series),", vol. 2021, no. Jun 8,.

[5] Anonymous "Carmen Sandiego | Netflix Official Site,", vol. 2021, no. Jun 8,.





# Rust College Center for Cybersecurity

The Rust College Center for Cybersecurity provides a computer science program with a concentration in cybersecurity (i.e., defense and operations). Students and faculty are provided opportunities to collaborate on interdisciplinary research and perform outreach with other institutions.



