PROJECT TRAINING WORKSHOP

TECHNICAL – PROGRAMMING BASICS

RAAFI
Builds You Best

UNDERSTANDING PROGRAMMING

- **What is Programming?**
  - **Programming is a way to “instruct the computer to perform various tasks”.**
- **What is Programming Language?**
  - **Computers understand instructions that are written in a specific syntactical form called a programming language.**
  - **A programming language provides a way for a programmer to express a task so that it could be understood and executed by a computer**
- **Why do we need to learn concepts of programming languages?**
  - **Increased ability to express ideas**
  - **Improved background for choosing appropriate languages**
  - **Increased ability to learn new languages**
  - **Better understanding of significance of implementation**
  - **Overall advancement of computing**
- **What would be the important aspects of any programming language?**

Readability	Writability
Reliability	Cost

UNDERSTANDING PROGRAMMING

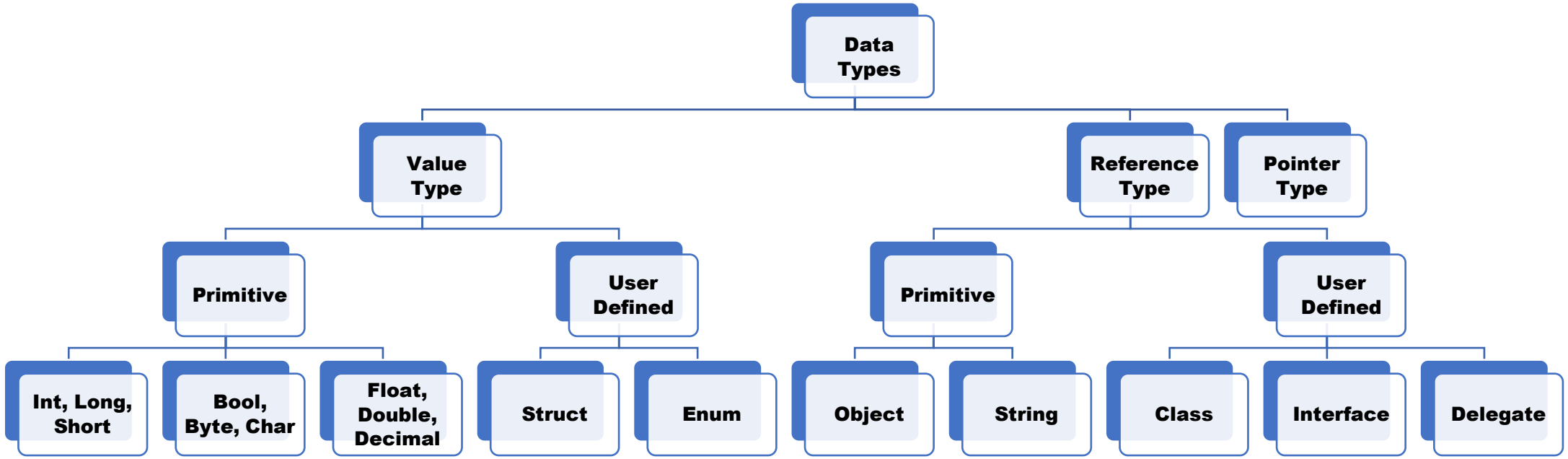
- **What are various types of programming language?**

Imperative – C, PASCAL	Functional – LISP, JavaScript	Logic - Prolog
Object Oriented – Java, C++	Markup – HTML, XML	

- **What are various Implementation methods of programming languages?**
  - **Compilation – Programs are translated into machine language**
  - **Pure Interpretation – Programs are interpreted by another program known as an interpreter**
  - **Hybrid Implementation Systems – A compromise between compilers and pure interpreters**
  - **Just In Time Implementation Systems**
    - **Initially translate programs to an intermediate language**
    - **Then compile intermediate language into machine code**
    - **Machine code version is kept for subsequent calls JAVA, .NET**
- **What are key concepts of programming language?**
  - **Data Types**
  - **Variables, Constants and Operators**
  - **Expressions, Statements and Control Structures**
  - **Sub programs and Blocks**
  - **Abstraction and Concurrency**
  - **Exception Handling**

UNDERSTANDING PROGRAMMING

- **What is Data Type?**
  - defines a collection of data objects and a set of predefined operations on those objects.
- **What are different types of Data Types?**



UNDERSTANDING PROGRAMMING

- **What is Variable?**
  - A variable is a named unit of data that is assigned a value. In other words, It is a memory location used to store a data value.
  - Ex: <data_type> <variable_list>
- **What is Constant?**
  - Const or constant is data or a value that does not change in a specified amount of time, unlike a variable.
  - Ex: const <data_type> <constant_name> = value;
- **What is Declaration, Initialization and Assignment?**
  - Declaration: Identifying the Name and Data Type of your data unit. Ex: int numberOfUnits;
  - Initialization: Giving the initial value for the data unit. Ex: int numberOfUnits = 5;
  - Assignment: Overriding the old value of your data unit with new. Ex: numberOfUnits = 10;
- **What is Operator and its Types?**
  - An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations
  - Types
    - Arithmetic Ex: + , - , * , /
    - Relational Ex: < , > , ==, !=, >= , <=
    - Logical Ex: && , || , !
    - Bitwise Ex: & , | , ^ , ~ , << , >>
    - Assignment Ex: = , += , -= , *=
    - Misc Ex: sizeof(), typeof(), is, as, ?:

UNDERSTANDING PROGRAMMING

- **What is Expression and its Types?**
  - **An expression is a combination of operators and operands which reduces to a single value. An operation is performed on a data item which is called an operand. An operator indicates an operation to be performed on data. Ex: sum = a * b + 10**

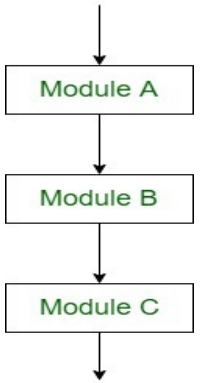
Variable	Operator	Variable	Operator	Variable	Operator	Constant
Sum	=	a	*	b	+	10

  - **Types**
    - **Constant Expressions Ex: 5, 10 + 5 / 6.0**
    - **Integral Expressions Ex: x, x * y**
    - **Floating Expressions Ex: x + 10.75**
    - **Relational Expressions Ex: x <= y, age >= 10**
    - **Logical Expressions Ex: x > y && x == 10, x == 10 || y == 5**
    - **Pointer Expressions Ex: &x**
    - **Bitwise Expressions Ex: x << 3, m | n**
    - **Compound Expressions : two or more of above expressions combined**

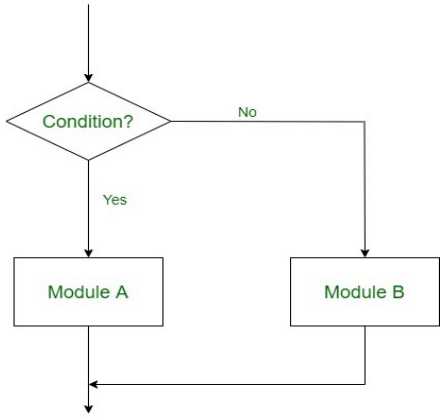
- **What is Statement and its Types?**
  - **The actions that a program takes are expressed in statements. A statement can consist of a single line of code that ends in a semicolon, or a series of single-line statements in a block**
  - **Common action includes,**
    - **declaring variables**
    - **assigning values**
    - **calling methods**
    - **looping through collections**
    - **branching to one or another block of code**

UNDERSTANDING PROGRAMMING

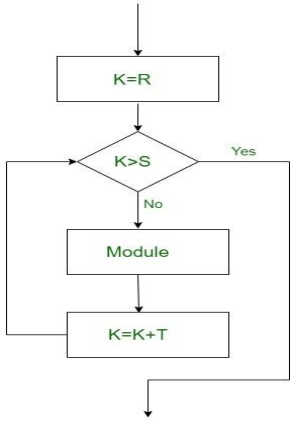
- **What is Control Structure and its Types?**
  - **Control Structures are just a way to specify flow of control in programs.**
  - **Types**
    - **Sequential Flow (Sequence Logic)**
    - **Conditional Flow (Selection Logic)**
    - **Repetitive Flow (Iteration Logic)**



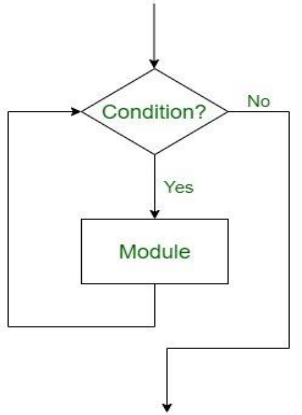
Sequential Flow (Sequence Logic)



Conditional Flow (Selection Logic)



Repetitive Flow (Iteration Logic) - FOR



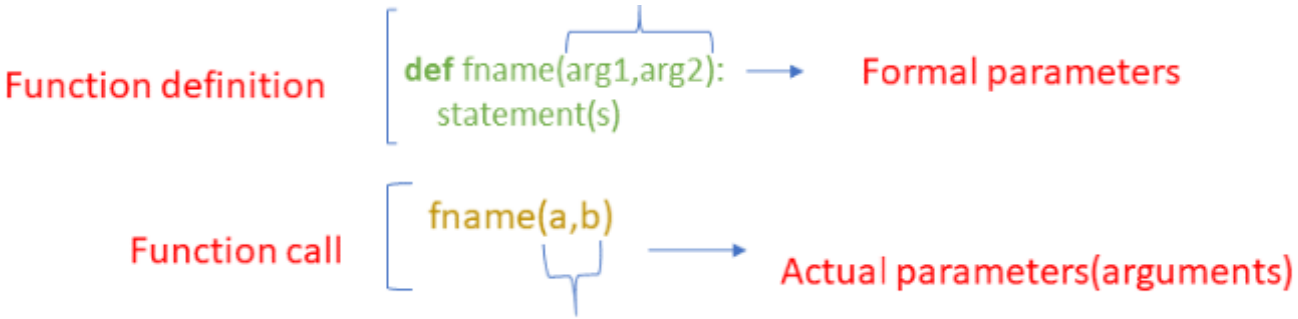
Repetitive Flow (Iteration Logic) - WHILE

UNDERSTANDING PROGRAMMING

- **What is Subprogram?**
  - **A Subprogram is a program inside any larger program that can be reused any number of times.**

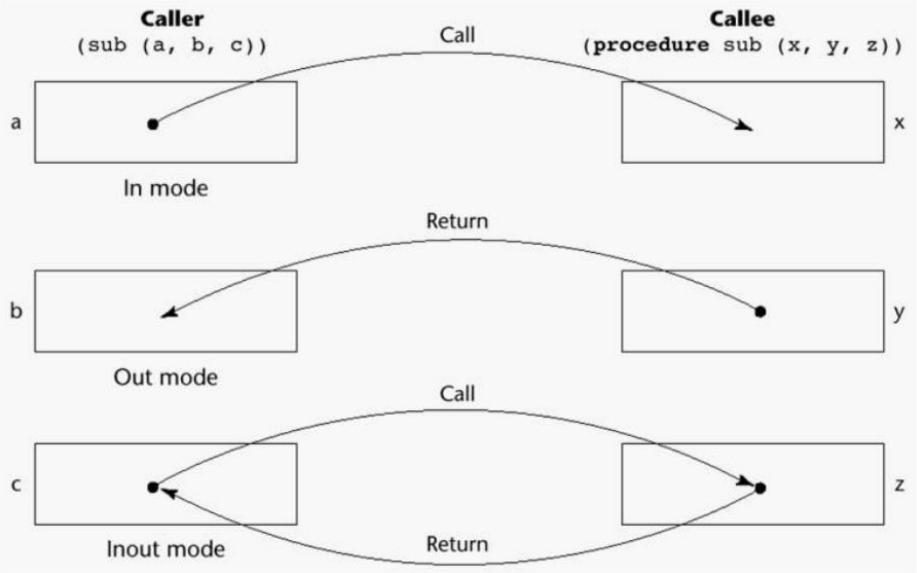
- **What are two categories of subprograms?**
  - **Procedure - A procedure is used to perform certain task in order.**
  - **Function (or method) - A function is used to calculate result using given inputs.**

- **What is function Parameter and Argument?**
  - **Argument - a value passed to a function when the function is called.**
  - **Parameter - a placeholder for the argument during function execution**



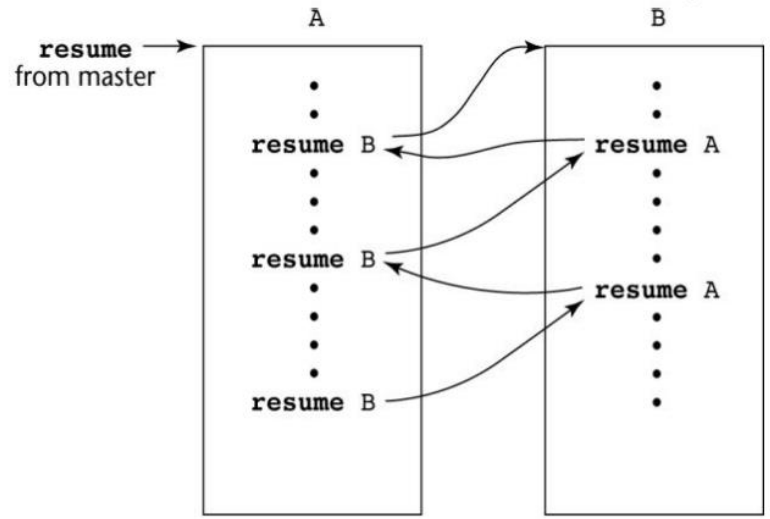
UNDERSTANDING PROGRAMMING

- **What are different ways of parameter passing?**
  - **Pass-by-value (IN mode)**
  - **Pass-by-result (OUT mode)**
  - **Pass-by-value-result (INOUT mode)**
  - **Pass-by-reference (INOUT mode)**
  - **Pass-by-name (INOUT mode)**



UNDERSTANDING PROGRAMMING

- **What is co-routine?**
  - **A coroutine is a subprogram that has multiple entries and controls them itself.**



- **What is Block?**
  - **Blocks are user-specified local scopes for variables ex: { // your code here }**

- **What is Scope and Lifetime?**
  - **Lifetime - Refers to how long or when the variable is valid (i.e. how long will it retain its value for).**
  - **Scope - Refers to where the variable can be accessed.**

UNDERSTANDING PROGRAMMING

- **What is Abstraction?**
  - The purposeful suppression or hiding of some detail of a process or artifact in order to bring out more clearly other aspects details or structure

- **What are different types of Abstraction?**
  - Procedural Abstraction - It includes series of the instructions having the specified functions.
  - Data Abstraction - It is set of data that specifies and describes a data object.
  - Control Abstraction - It is program control mechanism where interior details are not specified.

- **What is Concurrency?**
  - Concurrency is an ability of a program to do multiple things at the same time.

- **What is Exception, Exception Handling and Exception Handler?**
  - Exception - An exception is any unusual event, either erroneous or not, detectable by either hardware or software, that may require special processing.
  - Exception Handling - The special processing that may be required after detection of an exception is called exception handling.
  - Exception Handler - The exception handling code unit is called an exception handler

PROGRAMMING BASICS

- **ASSESSMENT**
  - **Download any Application/Project code and apply your learnings**