INTRODUCTION

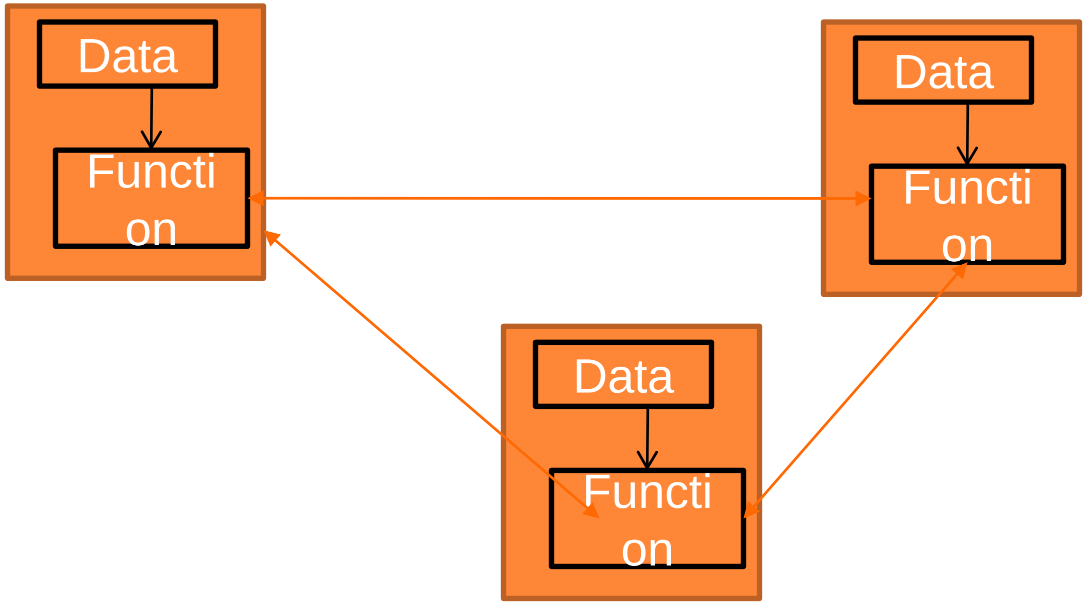
1. Object –Oriented Programming Languages
2. Functional Programming Languages
3. Logic/Non Procedural Languages
4. Comparison b/w C & C++

3.Object –Oriented Programming Languages:

Salient features are:

- Programs are divided in to what is known as objects
- Data is hidden and can't be accessed by external function
- Objects may communicate through function
- New data and function can be easily added when needed
- Follows Bottom-Up approach in program design

Organization of Data and Function Concepts



3.Object –Oriented Programming Languages:

Fundamental Concepts:

- Object:

Objects are the basic run time entities in an object-oriented system. They may represent a person, a bank account ,a place, a table of data or any item Programming problem is analyzed in terms of objects. Program objects should be chosen in such way that they should match with the real world objects.

- Class:

A class is a template or prototype from which objects are created, so it describes a collection of variables and methods .These methods can be accessible to the all other classes(public methods) or can have a restricted access(private methods)New class can be derived the parent class. So a class defines the abstract properties of a object including the objects properties as well as objects

3.Object –Oriented Programming Languages:

Fundamental Concepts:

Behavior(method, operation etc.) too

- **Data Abstraction and Encapsulation:**

The wrapping up of data and functions in to a single unit(called class) is known as encapsulation. Infect it is the most striking feature of the class. The data is not a accessible to the outside world.

This isolation access of data from direct access by the program is known as **data hiding** or the **information hiding**

- **Inheritance:**

It is the process by which objects of the one class acquire the properties of objects of another class. It supports the concepts of the inheritance provides the idea of reusability. It’s mean that we can add additional features to an existing class without modifying it. This is possible by the deriving a new class from the existing one.

3.Object –Oriented Programming Languages:

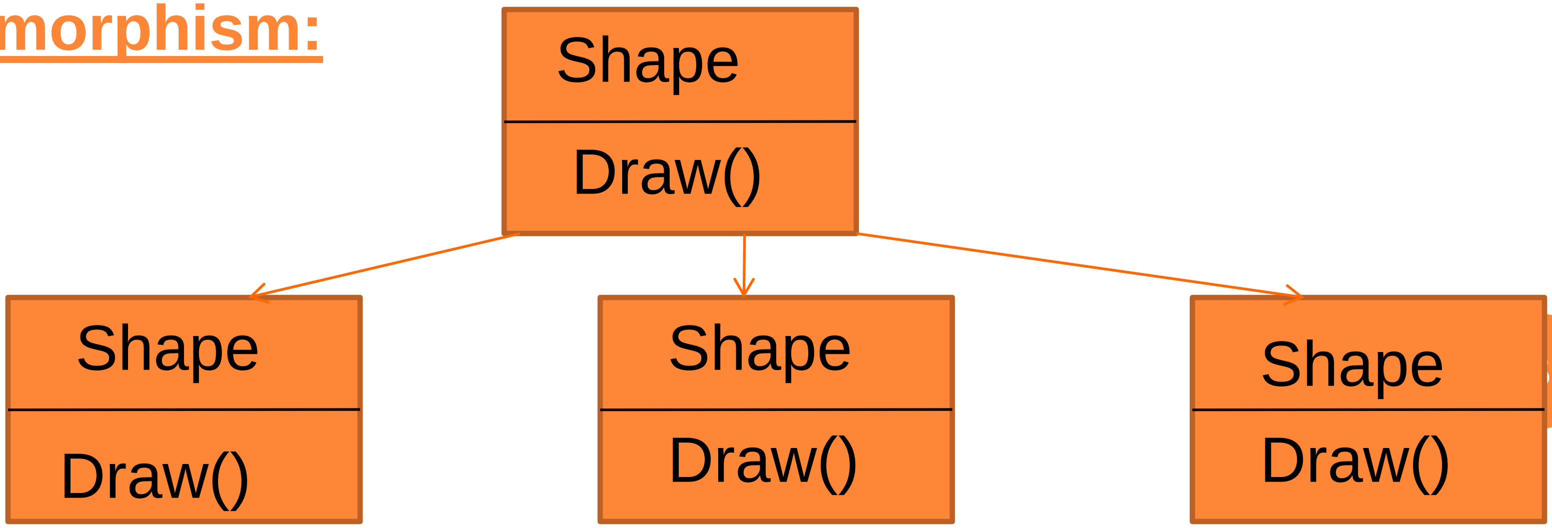
- **Inheritance:**

The new class will have the combined features of both classes.

- **Dynamic Binding:**

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding also known as the late binding means that code associated with a given procedure call is not known until the time of the call at the run time. It is associated with the polymorphism and inheritance.

- **Polymorphism:**



3.Object –Oriented Programming Languages:

•**Polymorphism:**

Polymorphism is another important OOP concept. It's mean that ability to take more than one form. An operation may exhibit different behavior at different instances. The behavior depends on the types of the data used in the operation. Using a single function name to perform the different types of task is known as the **Function Overloading**

• **Message Passing:**

The process of programming in an object-oriented language involves the following steps:

1. Create class that define the objects and their behavior.
2. Creating objects from the class definition
3. Establishing the communication among objects

Objects communicate with each other by sending and receiving information.

3.Object –Oriented Programming Languages:

•**Advantages**

•Through inheritance, we can eliminate the redundant code and extend the use of existing classes.

•Higher productivity through standard working modules.

• Data is much more secure by the means of data hiding.

•It is possible to have multiple instances of an object to co-exist without any interference.

•Upgrade easily from small to large system

•Software complexity can be easily managed.

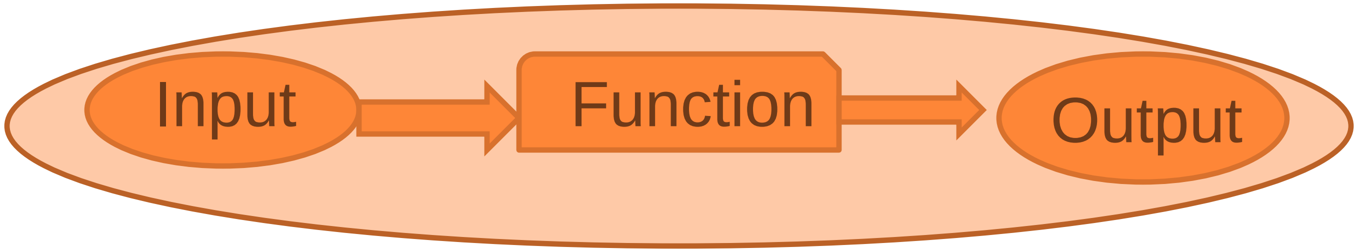
•It is mostly suitable for larger software project.

Languages comes under this category:

1. **Java**
2. **Visual Basic**
3. **C#**
4. **C++**

4.Functional Programming Languages:

It is so called as a program consists entirely of the functions. The main program itself is written as function which receives program's input as its argument and delivers the program's output as its result. This style is totally reverse from the OOPS. For example in OOPS Data name is first where as in the function programming the function name is first.



Concepts:

1. **Higher order Function:**

Function are high order when they can take other function as an arguments and returns them as a results. High order function enable currying, a technique in which a function is applied to its arguments one at a time, with each application returning a new(higher-order)function that accepts the next argument.

4.Functional Programming Languages:

Concepts:

2.Pure Function:

Functional programming involves writing programs using pure function are high order when they can take other function as an arguments and returns them as a results. High order function enable currying, a technique in which a function is applied to its arguments one at a time, with each application returning a new(higher-order)function that accepts the next argument. This allows a very clean, very high-level,very concise programming model, which is also:

- **Quick to write**
- **Easy to reason about**
- **Easy to modify/debug**
- **Easy to parallelize**

3.Recursion:

Recursive function invokes themselves allowing an operation to be performed over and over. Recursion may require

4.FUNCTIONAL PROGRAMMING LANGUAGES:

Concepts:

3.Recursion:

Maintaining a stack.

4.Strict,Non-Strict and lazy Evaluation:

Functional languages can be categorized by if they use strict or non-strict evaluation. Means that how function arguments are processed when an expression is being evaluated. The need for a more efficient form of Non-strict evaluation led to the development of lazy evaluation.

Advantages:

- Functional program contains no assignment statements, so variables once given a value never change.
- These program have no side effects.
- Take less time to develop & test an application as compared to the procedural style languages
- Suitable to concurrent environment

4.Functional Programming Languages:

Languages that comes under this category:

- PROLOG
- ML
- LISP

5. Logic/Non Procedural Languages:

Languages used for logic programming is known as the declarative language as because the program written in the form of declaration rather than the assignment and control flow statements.

One of the required characteristics of logic programming language is their semantics that is called declarative semantics. Logical programming languages/non-procedural language is one in which we specify **what needs to be computed** but not **"How it is to be done."**

That is, one specifies:

- The set of objects involved in computation
- The relationships that hold between them
- The constraints that must hold for the problem to be solved

Logical programming languages provides one kind of rule based programming environment

- Prolog is the only widely used programming laguages.

5. Logical/Non Procedural Languages:

Advantages of PROLOG:

- It is simple language
- Small ,fast
- Easy to write good compliers for it

Disadvantages:

- It has a fixed control strategy
- It has a strong procedural aspect
- Limited support Concurrency

Advantages of Logical Programming Languages:

- These languages are free to worry about explicit control
- Data can be represent in two ways extensionally or intentionally
- best suited to solve the complex ideas
- High level these languages focus on computation logic not on its method

5. Logical/Non Procedural Languages:

Logical Languages Disadvantages:

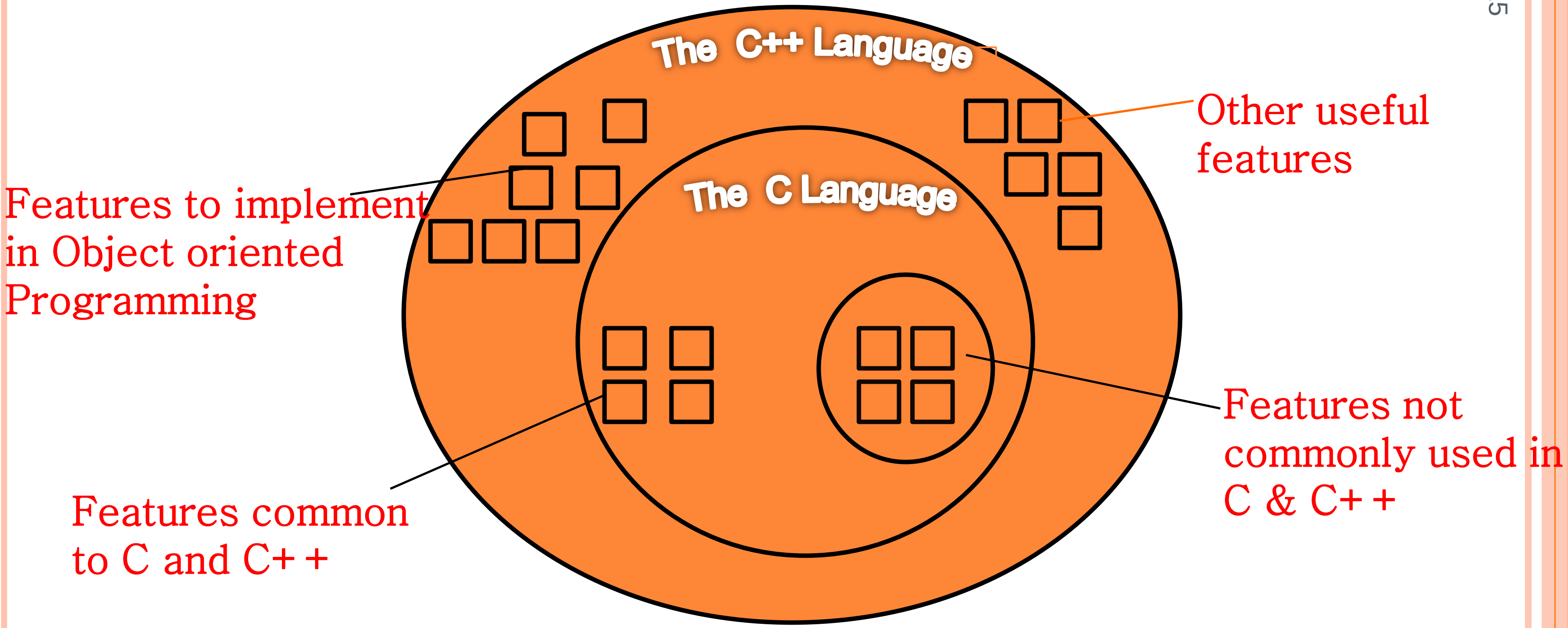
1. These languages have certain restriction

Application:

1. Relational Database Management System
2. Expert System
3. Natural Language Processing
4. Education

Comparison b/w C & C++

C++ is derived from the C language. Or we state that C++ is a super set of C. Almost every correct statement in C is also a correct statement in C++ but reverse is not true.


The C++ Language
The C Language
Other useful features
Features to implement in Object oriented Programming
Features not commonly used in C & C++
Features common to C and C++

Comparison b/w C & C++

C	C++
It is a procedural language	C++ is a object oriented
In C we use #include<stdio.h> as a inclusion file	#include<iostream.h> as a inclusion file
We can implicitly assign a void* to any other data type.	C++ we can't do this
In c you are not forced to declare the function before used it	In c++ you are strictly enforced to declare all function before used
In c you must manually as the statement ‘return0’ at the end of main	In c++ it will be provided automatically
C doesn’t have the reference variables and name space	C++ have reference variables and namespace

Comparison b/w C & C++

C	C++
You can't overload a function in C	You can overload a function in C++
Structure in C can not have function	In C++ structure can have function
C recognizes only the first 32 character in a name	C++ places no limit on name's length
C requires all the variables to be defined at the beginning of a scope	C++ don't require this requirement
Variables can't be initialized at the run-time	Variables can be initialize at the run-time
Input and output statement is present printf() and scanf()	Specific input and output statement are present Cin & Cout

Comparison b/w C & C++

C	C++
In c language the function malloc() and free() are used for allocation	In c++ new() and delete() function are used
There is no inline function in c	There is concept of inline function
There is no mechanism for exception handling.	C++ support exception handling
No virtual function in c	There is virtual function