OBJECT ORIENTED PROGRAMMING IN C++

BY CODE WITH HARRY

WHY OOPS?

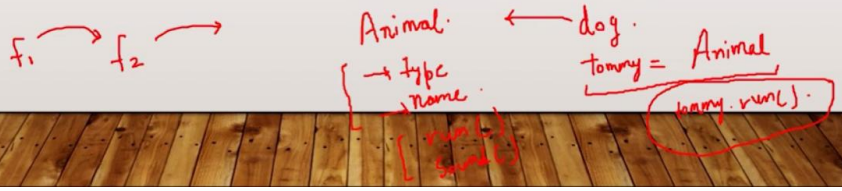
Initial name of “C++” is “C with classes”

- C++ language was designed with the main intention of adding object-oriented features to C language.
- As the size of the program increases, readability, maintainability and bug-free nature of programs decreases.
- This was the major problem with languages like C which relied upon functions or procedures (hence the name procedural programming language)
- As a result, the possibility of not addressing the problem in an effective manner was high.
- Also, as data was almost neglected, data security was easily compromised.
- Using classes solves this problem by modelling program as a real world scenario

C with classes.

WHY OOPS?

- C++ language was designed with the main intention of adding object-oriented features to C language.
- As the size of the program increases, readability, maintainability and bug-free nature of programs decreases.
- This was the major problem with languages like C which relied upon functions or procedures (hence the name procedural programming language)
- As a result, the possibility of not addressing the problem in an effective manner was high.
- Also, as data was almost neglected, data security was easily compromised.
- Using classes solves this problem by modelling program as a real world scenario

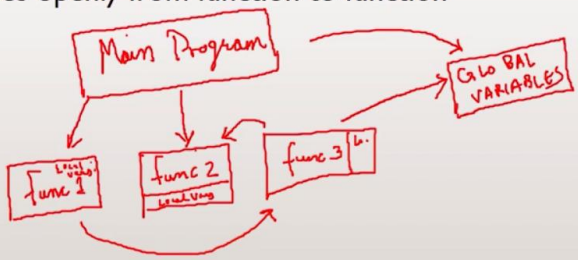


PROCEDURE ORIENTED PROGRAMMING

- Consists of writing a set of instructions for the computer to follow
- Main focus is on functions and not on flow of data
- Functions can either use local or global data
- Data moves openly from function to function

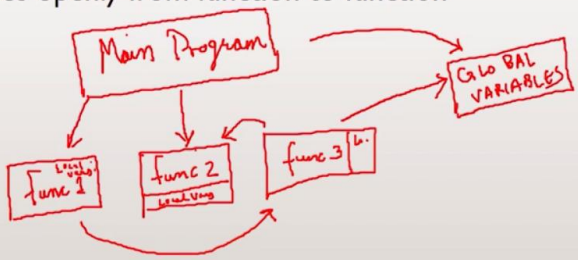
PROCEDURE ORIENTED PROGRAMMING

- Consists of writing a set of instructions for the computer to follow
- Main focus is on functions and not on flow of data
- Functions can either use local or global data
- Data moves openly from function to function



PROCEDURE ORIENTED PROGRAMMING

- Consists of writing a set of instructions for the computer to follow
- Main focus is on functions and not on flow of data
- Functions can either use local or global data
- Data moves openly from function to function



OOP

OBJECT ORIENTED PROGRAMMING

- Works on the concept of classes and objects
- A class is a template to create objects
- Treats data as a critical element
- Decomposes the problem in objects and builds data and functions around the objects

Animal

dog1 →

Cat1 → cat.sound='Meow'

Cat2

⋮

BASIC CONCEPTS IN OBJECT ORIENTED PROGRAMMING

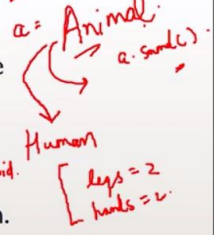
Abstraction → Separation of concerns

- Classes – Basic template for creating objects.
- Objects – Basic run time entities.
- Data Abstraction & Encapsulation – Wrapping data and functions into single unit.
- Inheritance – Properties of one class can be inherited into others.
- Polymorphism – ability to take more than one forms.
- Dynamic Binding – code which will execute is not known until the program. runs.
- Message Passing – Object.message(Information) call format.

"Data binding" is also known as "Late binding"

BASIC CONCEPTS IN OBJECT ORIENTED PROGRAMMING

- Classes – Basic template for creating objects.
- Objects – Basic run time entities.
- Data Abstraction & Encapsulation – Wrapping data and functions into single unit.
- Inheritance – Properties of one class can be inherited into others.
- Polymorphism – ability to take more than one forms.
- Dynamic Binding – code which will execute is not known until the program. runs.
- Message Passing – Object.message(Information) call format.



Volume (r, h) = Cyl.
lxbx = cuboid.

a = Thing b = Animal c = Human (child)

BENEFITS OF OBJECT ORIENTED PROGRAMMING

- Better code reusability using objects and Inheritance.
- Principle of data hiding helps build secure systems.
- Multiple objects can co-exist without any interference.
- Software complexity can be easily managed.