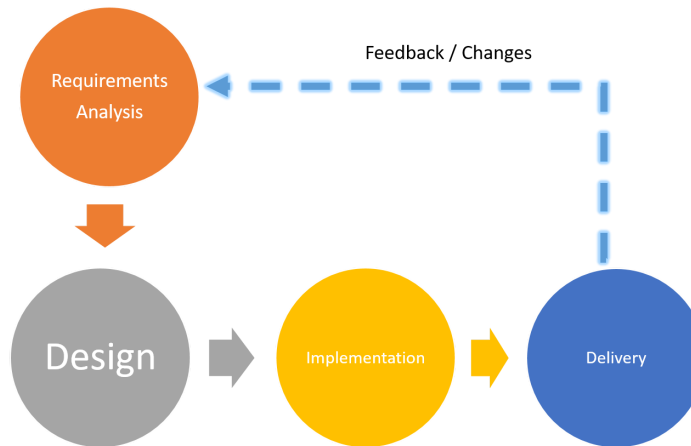


# Software Development Process

## Problem Solving Process

- In this section, we will be discussing a general approach to problem-solving.
- Let us discuss the steps that are followed in general to solve any problem

### Problem Solving Process

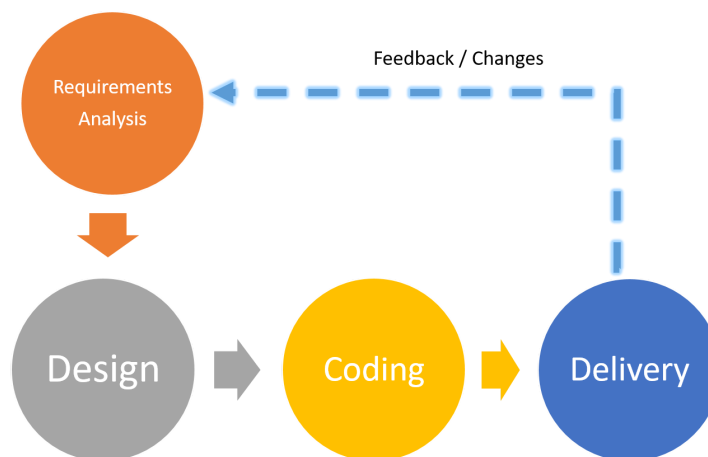


1. First, we focus on the requirements of the problem.
  2. Next, on the basis of understanding of requirements, we develop a design or prototype of the solution.
  3. Finally, we convert the prototype to the real implementation and deliver it to the customer.
- If the customer, after testing the solution, suggests a change, then the whole process is followed again to reflect that change.
  - Let's understand this with the help of the previously discussed examples of house construction.
  - This is how the house is constructed:
    1. Initially, all the requirements such as size, area, and no. of the floor are documented.
    2. Then a map is created considering these requirements.
    3. The details on the map are followed to build real houses.
    4. If the owner finds out some flaw after the house is built, he will have to get the new map created with updated requirements, and then only the changes will be made.
  - The same is the case with car manufacturing:
    - Requirements to car design.
    - Then car design to real car

# Software Development Process

- We saw the general problem-solving process which consist of 4 steps:
  1. Requirement Analysis
  2. Design
  3. Implementation
  4. Delivery
- The same set of steps is followed to develop any software. The only difference is that here implementation is the coding phase.

# Software Development Process

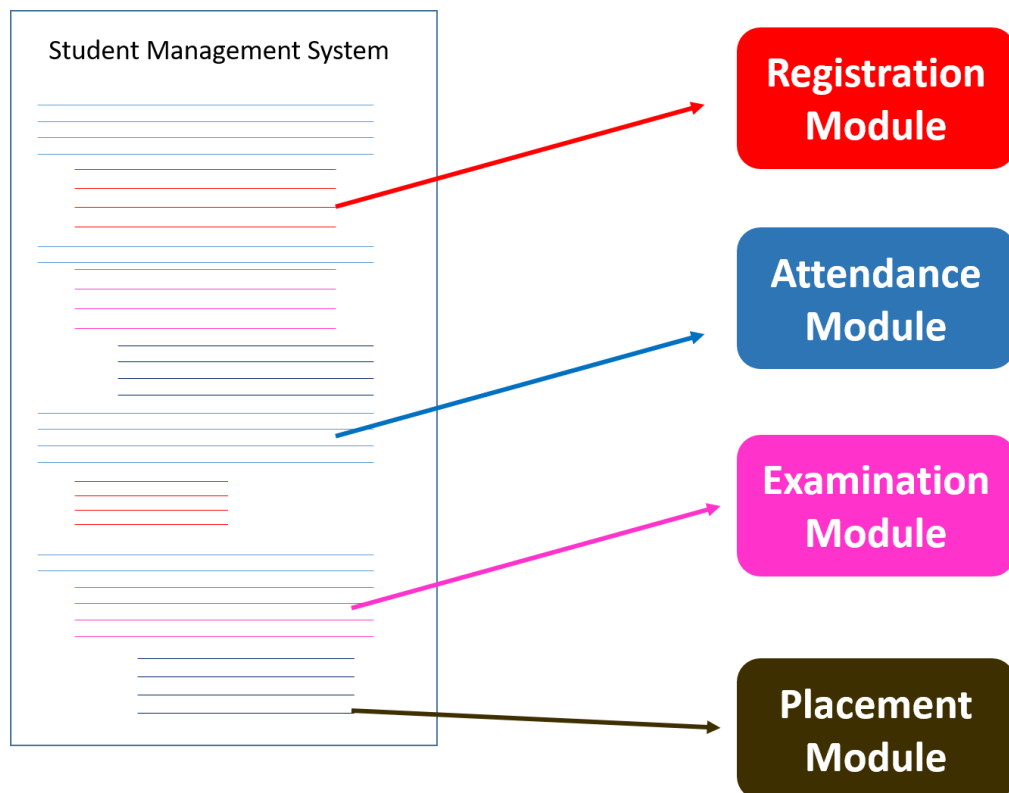


- There are **N** number of programming languages in the market that can be used to code software.
- You can pick any one of the languages that fits into your portfolio.
- So the coding phase consists of selecting a programming language or framework and converting the software designs to executable code.



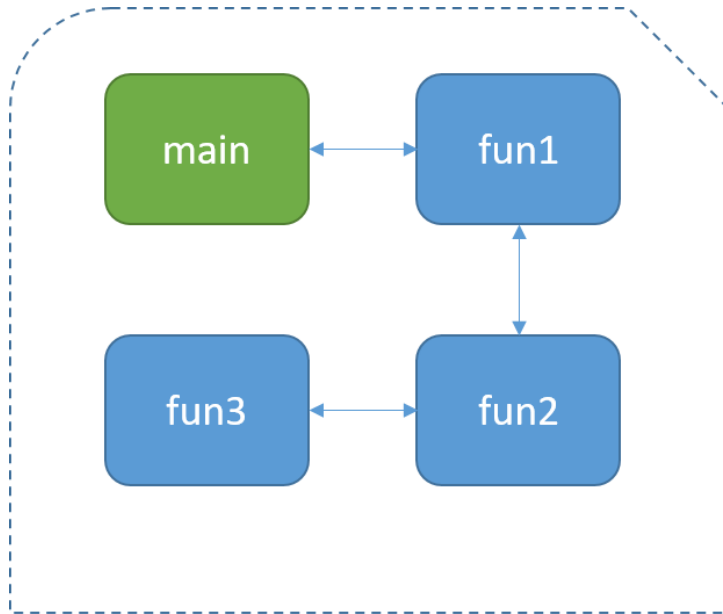
- Here, one thing that might have made you skeptical for sure is the DESIGN phase.
- It is easy to digest that a house can be visualized via a map.

- A car can be imagined with a design. But code, How will I visualize the code? How can we create designs for the code?
- I will help you to understand this. You must agree that software can never be a single large program, It has to be divided into modules.

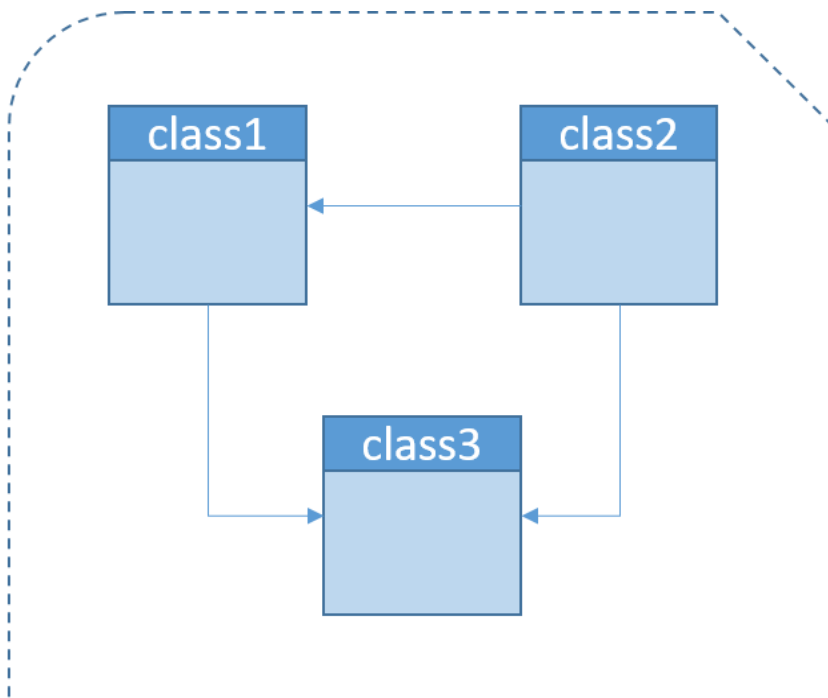


- Now, a software design or model can be a graphical representation of the Interaction of modules in the software.
- Now, a new question arises i.e., how to divide the software into a module. That means, how to identify the modules in our program.
- In the era of C and other procedural languages, the software was considered as the collection of functions or procedures.
- There used to be the main procedure that could interact with other procedures. At that time every procedure was considered as a module.
- So when the designs are created in which there is a module for every procedure, they are known as Procedural Designs.

- But, this procedural method had several drawbacks.



## Procedural Designs



## Object Oriented Designs

- The evolution of object oriented languages such as C++, and java, gave birth to object oriented designs where classes were considered as modules.
- In the object-oriented paradigm, a running program can be seen as a collection of objects collaborating to perform a given task.