

 Back To Course

LIVE BATCHES

 Learn 

Learn

 Quiz 

Learn

Quiz

We have combined Classroom and Theory tab and created a new Learn tab for easy access. You can access Classroom and Theory from the left panel.

 Problem Solving Process



Problem Solving Process

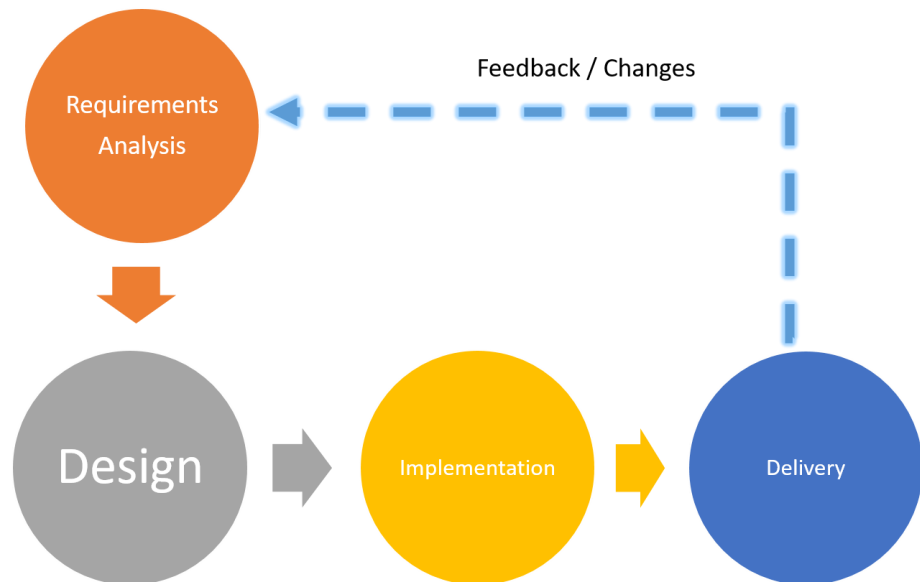
NOTES: Problem Solving Process



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.



Software Development Process



Software Development Process

NOTES: Software Development Process



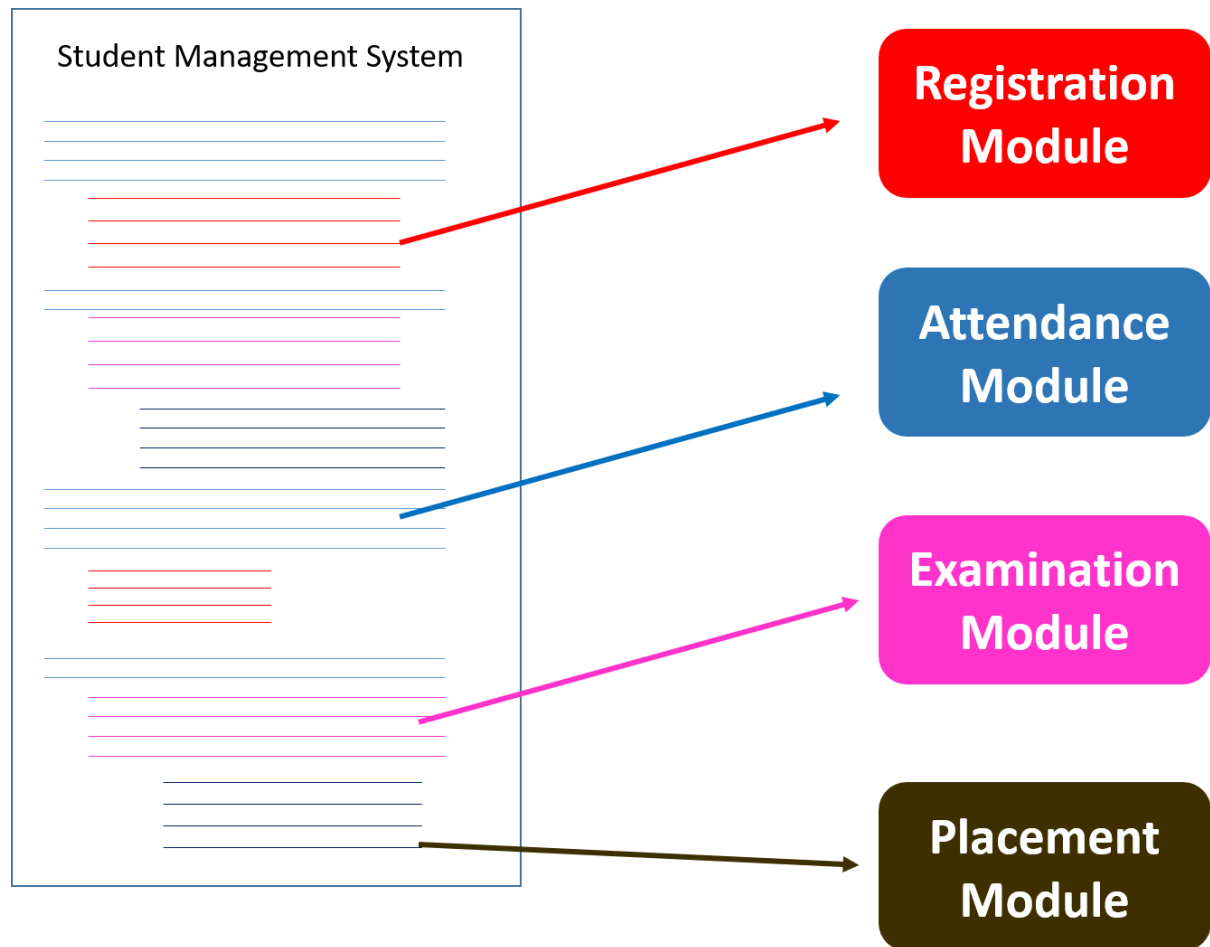
We saw the general problem-solving process which consist of 4 steps:

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

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.

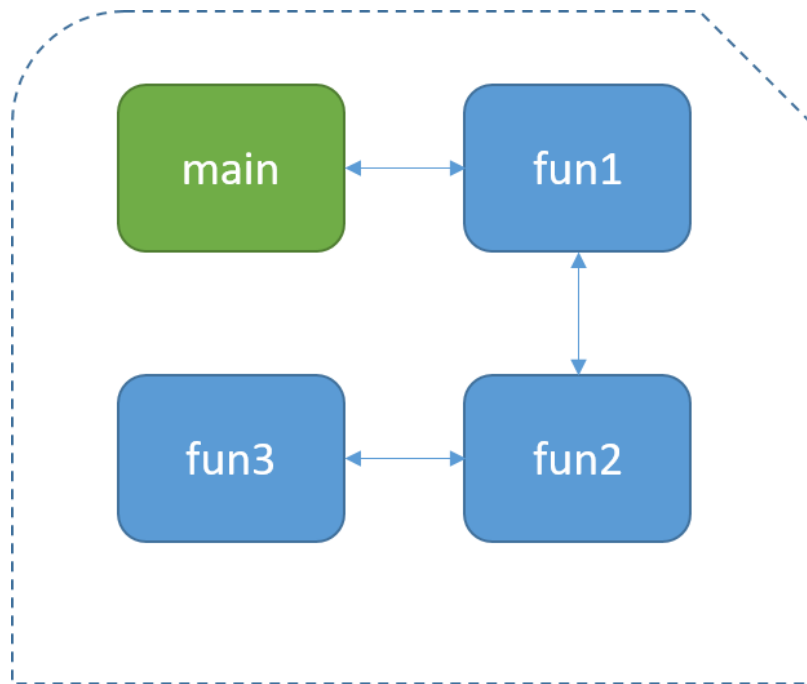
Now, a new question arises i.e., how to divide the software into a module. That means, how to identify the modules in our program.

In the era of C and other procedural languages, the software was considered as the collection of functions or procedures.

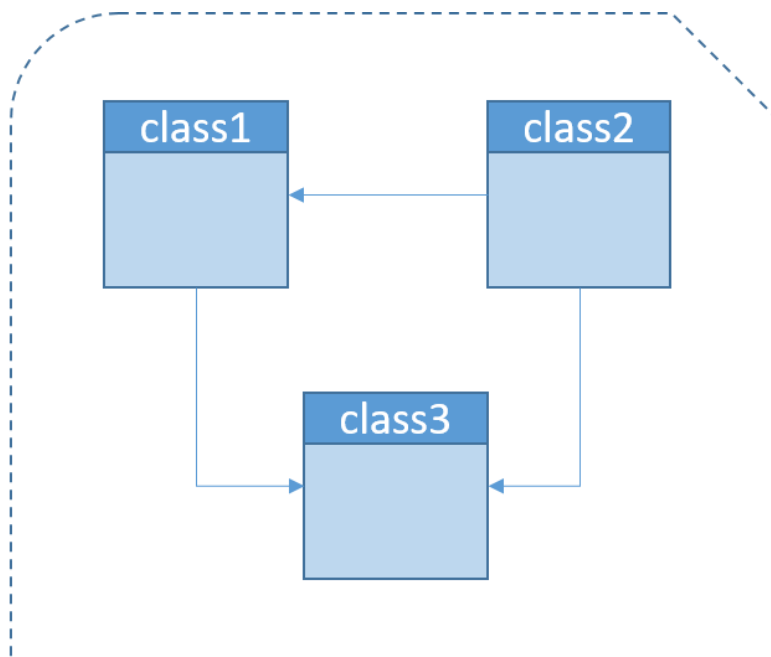
There used to be the main procedure that could interact with other procedures. At that time every procedure was considered as a module.

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.

 Report An Issue

If you are facing any issue on this page. Please let us know.

LIVE BATCHES



 5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh – 201305

 feedback@geeksforgeeks.org



Company

- About Us
- Careers
- Privacy Policy
- Contact Us
- Terms of Service

Learn

- Algorithms
- Data Structures
- Languages
- CS Subjects
- Video Tutorials

Practice

- Courses
- Company-wise
- Topic-wise
- How to begin?

Contribute

- Write an Article
- Write Interview Experience
- Internships
- Videos

@geeksforgeeks , All rights reserved