OBJECT ORIENTED PROGRAMMING

PROJECT BASED

Dr. Ann Alkazzaz

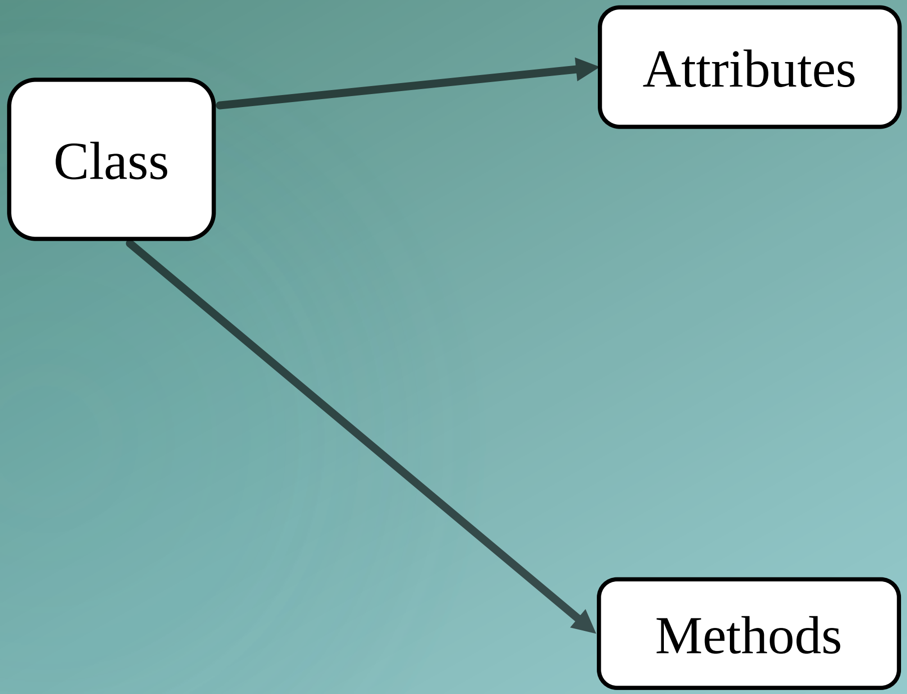
2nd semester (Lect1)

Quick review on OOP

In Python, everything is an object. Strings, Integers, Float, lists, dictionaries, functions, modules etc. are all objects.

- **Object and Class**

Class is an architecture of the object.



Class

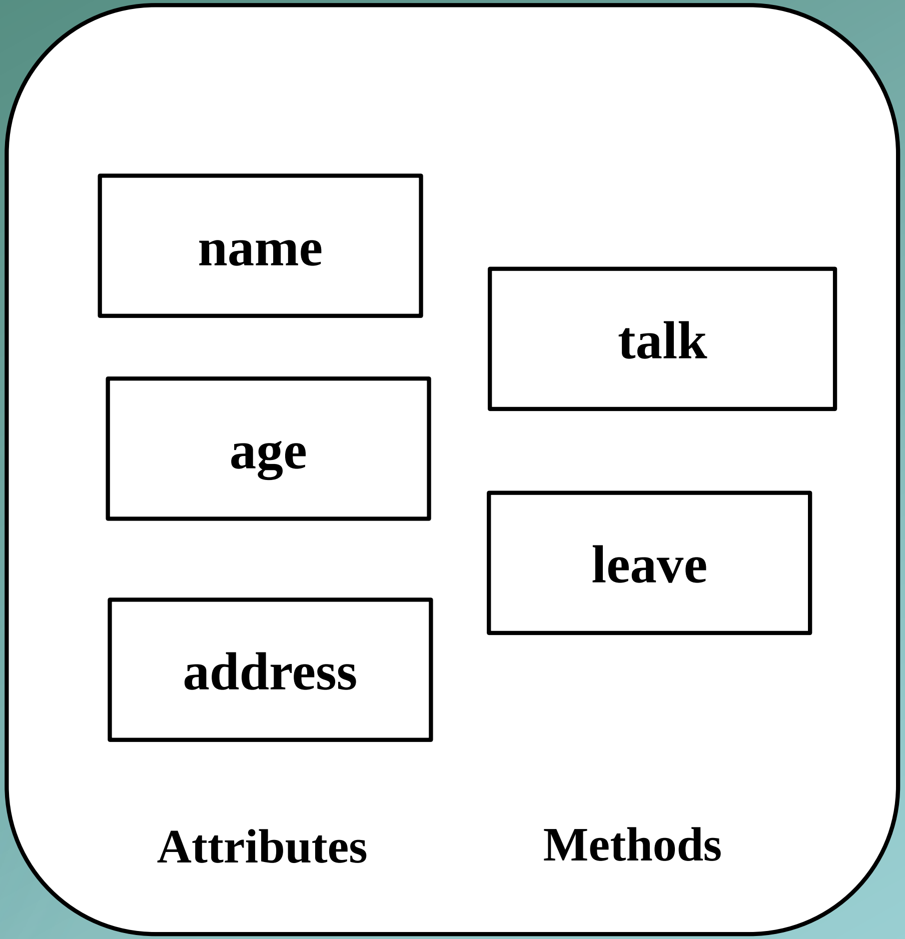
Attributes

Methods

- **The variables that belong to a particular object**
- To define the attributes we need to use constructor
- The **Constructor** is a function that gets called at the time of creating an object.
- All classes have a function called **__init__ ()** which is always executed when the class is being initiated.

Functions that operate on the attributes

Person Class example



name

age

address

talk

leave

Attributes

Methods

Person Class example

Now we will create new python file named **person.py**

```
class Person:
    def __init__(self,name,age, address):
        self.name = name
        self.age = age
        self.address = address

    def talk(self):
           print(f"hi everybody! this is {self.name} I am {self.age} years old, I live in {self.address} ")
    def leave(self):
          print(f"nice to meet you .Good bye!--{self.name}--Left ")
```

Introduction to Class Diagrams

- **Unified Modeling Language (UML)** is a standardized way to draw diagrams and pictures that help people understand how a software system works. It uses symbols and shapes to represent different parts of the system, like its structure and how it behaves.
- **Class diagram:** is a type of diagram in the Unified Modeling Language (UML) that represents the structure of a system by showing the classes, attributes, operations, and their relationships.
- **Purpose of Class Diagrams:**

The main job of a class diagram is to show us a simple picture of how a computer program is built. It helps everyone involved in making the program understand what each part does and how they connect.

Introduction to Class Diagrams

- **Components of a Class Diagram:**

Class diagrams have a few important parts:

1. **Class:** Represents a blueprint for objects in the system. It encapsulates attributes and methods that define the characteristics and behaviors of instances.
2. **Attributes:** Properties or characteristics of a class. Attributes provide details about the state of an object. For example, if we're talking about books, the attributes might include the book's title or the name of the author.
3. **Methods:** Functions or operations performed by a class. For our book example, methods might include actions like checking out a book or returning it.
4. **Relationships:** Describes associations between classes.This shows how things in the program work together.

How to design Class diagram

Library Management System class

This system manages all operations of Library Management System. It is central part of organization for which software is being designed.

1. **User Class** – Manages all user-related operations.
2. **Librarian Class** – Handles all librarian-related operations.
3. **Book Class** – It manages all operations of books. It is basic building block of system.
4. **Account Class** – It manages all operations of account.

How to design Class diagram

Attributes of Library Management System :

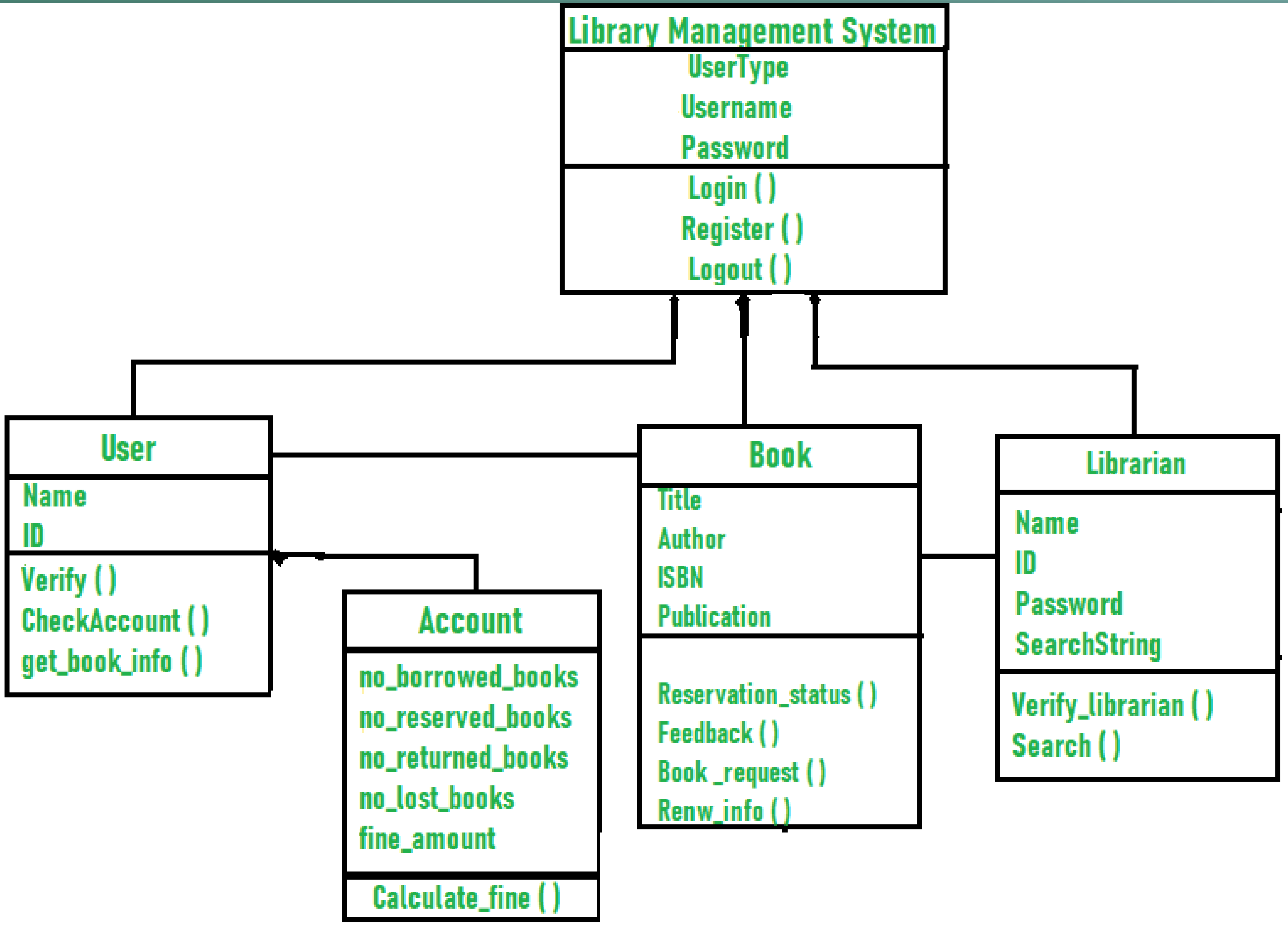
1. **Library Management System Attributes –** UserType, Username, Password
2. **User Attributes –** Name, Id
3. **Librarian Attributes –** Name, Id, Password.
4. **Book Attributes –** Title, Author, ISBN, Publication
5. **Account Attributes –** no_borrowed_books, no_reserved_books, no_returned_books, no_lost_books.

How to design Class diagram

Methods of Library Management System

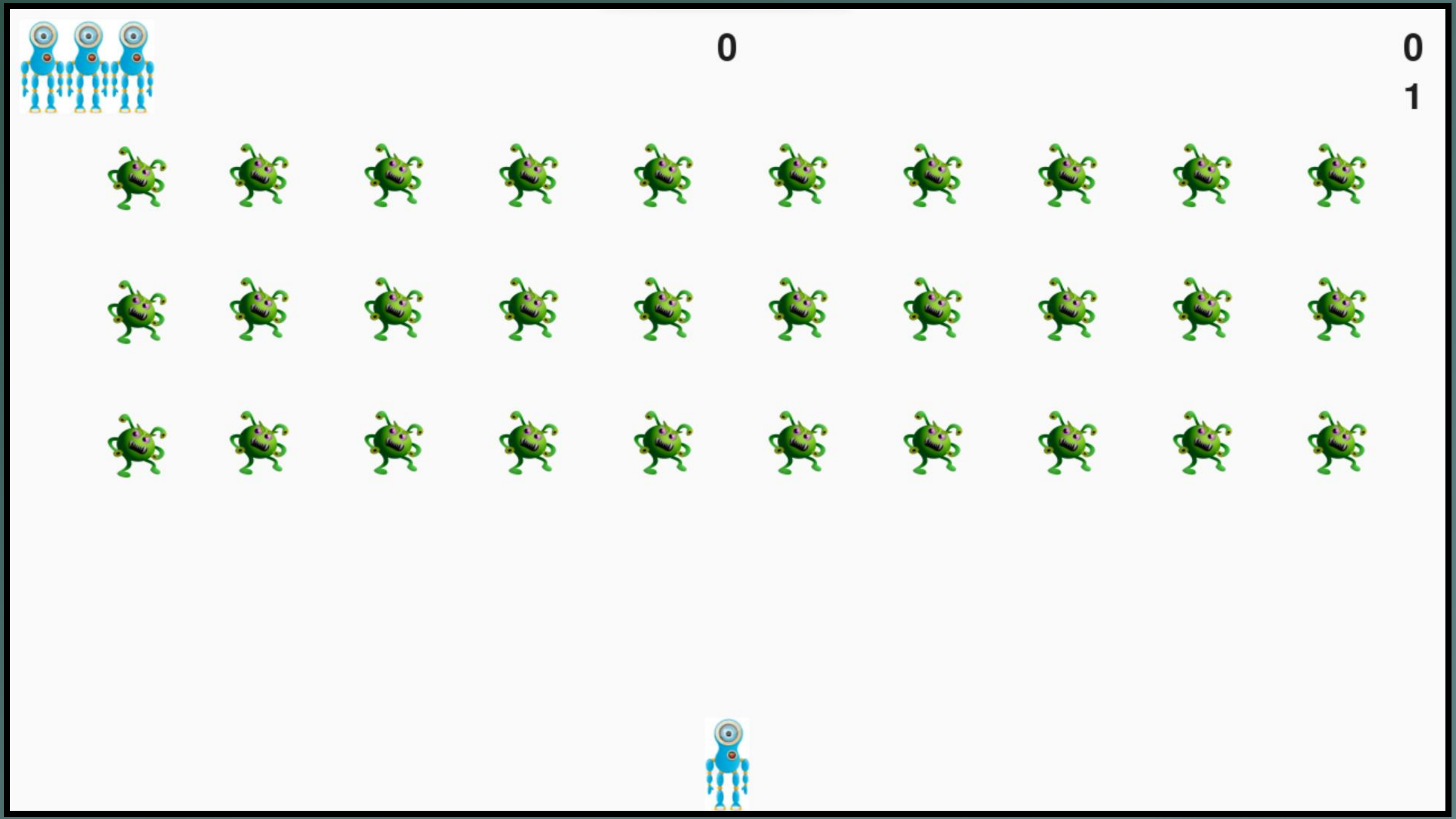
1. **Library Management System Methods –** Login(), Register(), Logout()
2. **User Methods –** Verify(), CheckAccount(), get_book_info()
3. **Librarian Methods –** Verify_librarian(), Search()
4. **Book Methods –** Reservation_status(), Feedback(), Book_request().

How to design Library Class diagram



CLASS DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM

Can you Construct a Class diagram for this game?



Our Project

- Making games is an ideal way to have fun while learning a language.
- writing a simple game will help you comprehend how professionals develop games.
- As you work through this project,
    - Run the code to understand the contribution of each code block to the overall gameplay.
    - Experiment with different values and settings to refine interactions in your games."

Our Project (Virus Defender)

❑ Description of the general gameplay

In ***Virus Defender***, the player controls a an antivirus robot that appears at the bottom center of the screen.
The robot shoots and destroys the viruses.

- If the player shoots all the viruses, the game will get faster than the before.
- If any virus hits the player's robot the player loses a robot.
- If the player loses three robots, the game ends