Introduction to Object-Oriented Approach

Objectives

- State the reasons for the complexity involved in the development of software
- Define the following terms
  - Objects
  - Classes
  - Messages
  - Methods
- Explain benefits of the object-oriented approach

Objectives (Contd.)

- State the significance of the activities involved in object-oriented analysis and design
- Create classes in C++

Introduction to Object-Oriented Approach

Complexity of the Software System

- Internal Complexity
  - Arises from the composition of a system itself
- External Complexity
  - Arises from the fact that users themselves have only a vague idea of how their system works and have difficulty in expressing their requirements

Introduction to Object-Oriented Approach

Reasons for the Complexity Involved in the Development of Software

- Difficulty in managing the software development process
  - How to represent real-life entities of problems in System Design.
  - How to ensure reusability and extensibility of modules.
  - How to improve software productivity and decrease software Cost .
  - How to manage time schedules ?
- Lack of standards for developing software
- Difficulty in predicting software behavior

Introduction to Object-Oriented Approach

Simplifying Complexity

- To build today's complex software it is just not enough to put together a sequence of programming statements and sets of procedures and modules.
- We need to incorporate sound construction techniques and program structures that are easy to comprehend ,implement and modify.
- Is done by breaking the system into its component parts and arranging them in a hierarchy

Introduction to Object-Oriented Approach

Just a Minute…

Jane has called a technician to repair her television. How would the technician deal with the complexity of the television?

Introduction to Object-Oriented Approach

Object

- Is an instance of a class that exhibits some well-defined behavior
- Are the basic run-time entities in an object-oriented system.
  - May represent a person,a place ,a bank account or any item that the program can handle.
- The\y interact by sending messages to one another

Introduction to Object-Oriented Approach

Characteristics of Objects

- State
  - Is indicated by a set of attributes and the values of these attributes
- Behavior
  - Is indicated by how an object acts and reacts
- Identity
  - Distinguishes the object from all other objects

Introduction to Object-Oriented Approach

Just a Minute…

Identify the possible attributes to define the state of the following objects:

- Tea cup
- Stereo tape-recorder

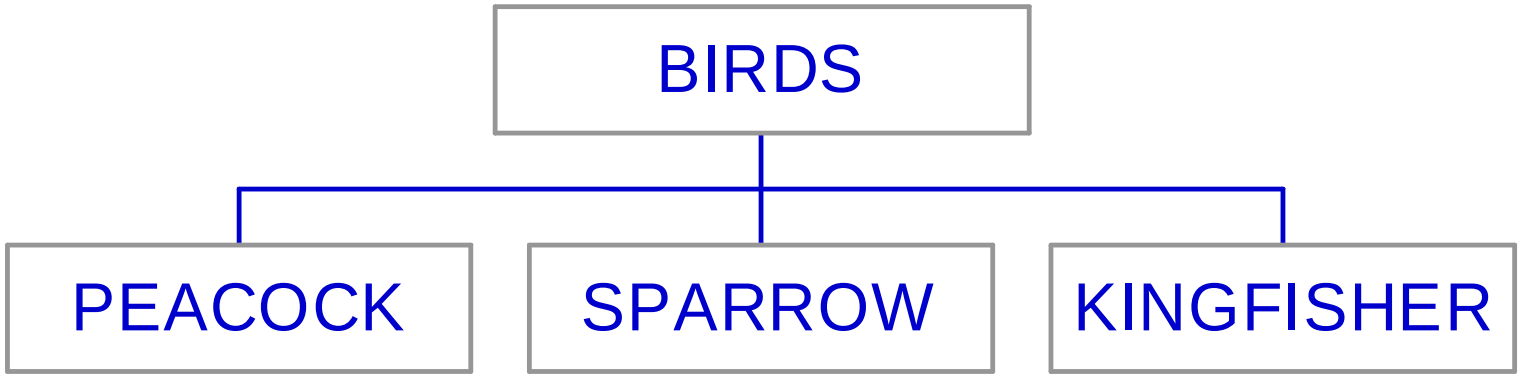
Introduction to Object-Oriented Approach

Classes

✉The entire set of data and code of an Object can be made a user-defined data type with the help of a Class.

✉ Define the attributes and behaviors of an object

✉ Example:



Introduction to Object-Oriented Approach

Messages and Methods

Messages:

- ✉ Are transmitted by one object to another
- ✉ Are transmitted as requests for an action to be taken
- ✉ Are accompanied by additional information needed to carry out the request

Messages and Methods (Contd.)

Methods:

- Are a set of actions taken by the receiver object in response to the request

Introduction to Object-Oriented Approach

Just a Minute…

Dr. James and Mr. Tom went to the railway station to book two tickets in the Flying express for 3rd December by AC 1st class. Identify the following:

a. The possible receiver of the message in this situation

b. The possible method that the receiver can use

Introduction to Object-Oriented Approach

Benefits of the Object-Oriented Approach

- Realistic modeling
  - Easy to use
  - Emphasis is on data rather than procedure.
  - Programs are divided into Objects
- Reusability
  - Saves time and cost
  - New data and function can be easily aded whenever necessary.

Introduction to Object-Oriented Approach

Just a Minute…

State whether the following situations demonstrate reusability:

a. Recycling paper

b. Pump reusability (same pump is used in a well and in a fuel station)

Benefits of Object-Oriented Approach (Contd.)

- Resilience to change
  - Easy to maintain
  - Parts of the system can be refined without any major change in other parts

Introduction to Object-Oriented Approach

Object-Oriented Analysis (OOA)

Analysis:

- Is a phase where users and developers get together and arrive at a common understanding of the system
- Requires the developer to concentrate on obtaining maximum possible information about the problem domain
- Results in one of the end products as specification of the function of the system

Introduction to Object-Oriented Approach

Object-Oriented Design (OOD)

Design:

- Generates the blueprint of the system that has to be implemented
- Involves identifying classes using desired technique.

Object-Oriented Design (OOD) (Contd.)

- Abbott's technique follows the listed steps:
  - Write English description of the problem
  - Underline nouns (nouns represent candidate classes)
  - Example:-

    There are several **counters** each managed by a single **salesperson** selling a specific **product**.The **customer** approaches any counter,depending on the product he/she wishes to purchase.The salePerson hands him/her the product and accepts **payment**.

Introduction to Object-Oriented Approach

Object-Oriented Programming (OOP)

- Is a way of writing programs
- Some applications built using OOP techniques are:
  - Computer-Aided Design (CAD)
  - Computer-Aided Manufacturing (CAM)
  - Artificial Intelligence (AI) and Expert Systems
  - Object-Oriented Databases

Introduction to Object-Oriented Approach

Just a Minute…

As a member of a team that is developing the billing system software for Diaz Telecommunications Inc., you have been assigned the task of creating a software module that accepts and displays customer details. Identify the class that you will create and the methods of the class.

Introduction to Object-Oriented Approach

Generations of Computer Languages

- First generation 1940s.
- Second generation 1950s.
- Third generation 1967.

Introduction to Object-Oriented Approach

Evolution of C++ as an Object Oriented Programming Language

- In the early 1980s, Bjarne Stroustrup developed the C++ language.
- C++ was originally known as 'C with classes'

Creating Classes in C++

Sample:

```
class Car
{
 public:
 void honk()
 {
  cout<<"BEEP BEEP!";
 }
};
```

Creating Classes in C++ (Contd.)

- The `class` keyword
  - Is used to declare a class

    Example:

```
class Car
{
...
};
```

Creating Classes in C++ (Contd.)

- Conventions for naming classes
  - Should be meaningful
  - Should ideally be a noun
  - First letter of every word should be in upper case
- Rules for naming classes
  - Must not contain any embedded space or symbol
  - Must begin with a letter, which may be followed by a sequence of letters or digits
  - Cannot be a keyword

Creating Classes in C++ (Contd.)

- Member functions
  - Are means of passing messages and responding to them
  - Are declared inside the class body

  Example:

```
class Car
{
void honk()
 {
  cout<<"BEEP BEEP!";
 }
};
```

Creating Classes in C++ (Contd.)

- The `cout` object
  - Is an instance of the pre-defined class, `ostream`
- The `endl` manipulator
  - Is a command that takes the cursor to the new line

Introduction to Object-Oriented Approach

Problem

As a member of a team that is developing the billing system software for Diaz Telecommunications Inc., you have been assigned the task of creating a software module that accepts and displays customer details. Declare the `Customer` class and the member functions. The member function to accept customer details should display the message “Accepting Customer Details”. Similarly, the member function to display customer details on the screen should display the message “Displaying Customer Details.”

Introduction to Object-Oriented Approach

Problem Statement

Solution:

```
class Customer
{
  void accept()
  {
  cout << “Accepting Customer Details” <<
  endl;
  }
  void display()
  {
  cout << “Displaying Customer Details” <<
  endl;
  }
};
```

Introduction to Object-Oriented Approach

Problem Statement

As a member of a team that is developing an automated booking system for the Railways, you have been assigned the task of creating a module that accepts the details of a passenger and checks whether the ticket has been confirmed or is in the waiting list. The module then prints the list of confirmed passengers. Declare a class `Ticket`, which consists of three member functions, `booking()`, `status()`, and `print()`.

Introduction to Object-Oriented Approach

Summary

- Complexity of software arises mainly due to four reasons:
  - Difficulty in understanding the intricacies and complexity of the system and its needs
  - Communication problems during development
  - Lack of standards for developing software
  - Difficulty in predicting software behavior
- One way of dealing with the complexity of software is to break down an application into its components and deal with each component separately

Summary (Contd.)

- The object-oriented approach views the systems as consisting of component objects and looks at the interactions between them
- An object is an entity that may have a physical boundary and is also characterized by the following:
  - State
  - Behavior
  - Identity
- A class consists of a set of objects that share a common structure and behavior
- If one object desires an action from another object, it sends a message to the second object

Summary (Contd.)

- The object that receives the message is called the receiver; the set of actions taken by the receiver constitutes the method
- The benefits of the object-oriented approach are:
  - Realistic modelling, hence it is easier to use
  - Reusability of code, hence it saves time and cost
  - Resilience to change, hence systems are easier to maintain
- In the stages of analysis and design, a model of the system is built

Summary (Contd.)

- The purpose of the model built during analysis and design is to help developers understand the reality that they are trying to imitate
- Bjarne Stroustrup developed the C++ language in the early 1980s
- The `cout` object is an instance of the class, `ostream`
- The class `ostream` is associated with the standard output device (screen)
- The output operator '<<' is used to direct a value to the standard output device