CS1529
Object Oriented Programming using JAVA

About the Course

- It is about object-oriented programming (OOP) principles
- Use of Java programming language to implement various aspect of OOP
- Get acquainted with the concept of Inheritance and Polymorphism as implemented in JAVA
- Learn to use the JAVA APIs, GUI components and to develop object-oriented based applications using JAVA Framework

Prerequisites

- Fundamentals of computer science
- C programming language
- Data structure and algorithms

Books

Text Book:

1. **Schildt, Herbert. Java. The Complete Reference. Ninth edition, McGraw-Hill Education, 2014.**

2. **E Balagurusamy, Programming with Java, McGraw Hill Education, 2019**

3. John R. Hubbard, Programming With Java, second edition, McGraw Hill, 2020.

References:

1. Sierra, Kathy, and Bert Bates. Head First Java. 2nd ed, O'Reilly, 2005.

2. Burd, Barry. Java for Dummies. Seventh edition, John Wiley & Sons, 2017.

3. Bloch, Joshua. Effective Java. Third edition, Addison-Wesley, 2018.

4. Loy, Marc, et al. Learning Java: An Introduction to Real-World Programming with Java. Fifth edition, O'Reilly, 2020.

Evaluation Criteria

- Mid Sem – 30 Marks
- End Sem – 50 Marks
- Teacher Assessment – 20 Marks
  - Classroom Assignment
  - Quiz
  - Attendance
  - Development Task

Relationship of Java with C

- Java is related to C++
- C++ is a direct descendant of C
- Much of the character of Java is inherited from these two languages
- Java derives its syntax from C
- Many of Java's object-oriented features were influenced by C++

Popular Java Development Tools

- Eclipse
- Intellij IDEA
- Visual Studio Code
- Apache Maven
- NetBean

Object-oriented Programming

- Object-oriented programming was developed because limitations were discovered in earlier approaches to programming

Procedural Languages

- Each statement in this language tells the computer to do something:
  - Get some input,
  - add these numbers,
  - divide by six,
  - display that output.
- A program in a procedural language is a **list of instructions.**
- When programs become larger, a single list of instructions becomes unwieldy.

Procedural Languages(2)

- Function was adopted as a way to make programs more comprehensible
  - also may be referred to as a subroutine, a subprogram, or a procedure in other languages
- A procedural program is **divided into functions**, and (ideally, at least) each function has a clearly defined purpose and a clearly defined interface to the other functions in the program.
- C, Pascal, FORTRAN, and similar languages are procedural languages.

Structured programming

- Grouping a number of functions together into a larger entity called a module (which is often a file)
- Dividing a program into functions and modules is one of the cornerstones of **structured programming**
- Structured programming influenced programming organization for several decades before object-oriented programming.

Problems with Structured Programming

- Increase in complexity as the program grows larger
- Functions have unrestricted access to global data
- Functions and data are unrelated.
- A change made in a global data item may necessitate rewriting all the functions that access that item.
- Procedural paradigm provides a poor model of the real world

Real-world modeling

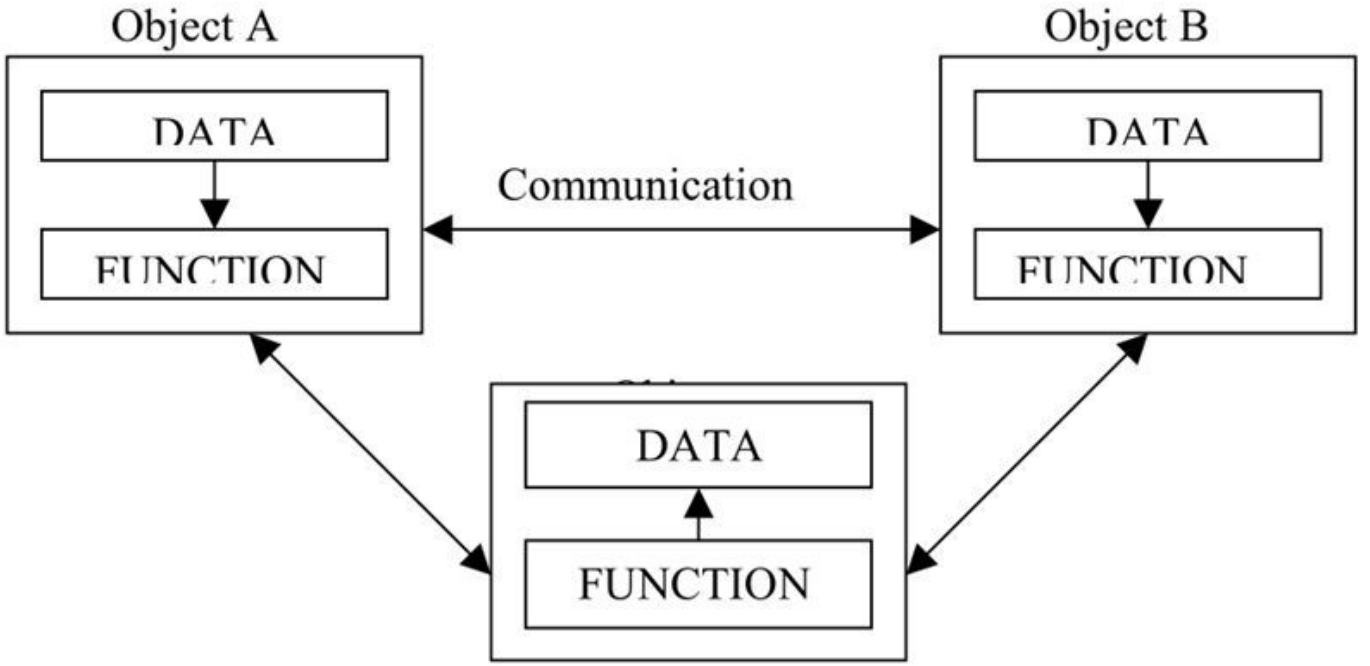
- In the physical world we deal with objects such as people, institution, and cars.
- These are neither just data and nor like functions.
- Complex real-world objects have both attributes and behavior.
- Attributes sometimes called characteristics in the real world are equivalent to data in a program: they have a certain specific values
- Behavior is something a real-world object is like a function: you call a function to do something (display the inventory, for example).
- So neither data nor functions, by themselves, model real-world objects effectively.

The Object-Oriented Approach

- The fundamental idea : “combine both data and the functions that operate on its data into a single unit”.
- Such a unit is called an object.
- An object’s functions, called member function, typically provide the only way to access its data.
- The data is hidden, so it is safe from accidental alteration.
- Data and its functions are said to be encapsulated into a single entity.
- Data encapsulation and data hiding are key terms in the description of object-oriented languages.
- This simplifies writing, debugging, and maintaining the program.

Object-oriented program

- An Object-oriented program typically consists of a number of objects, which communicate with each other by calling one another's member functions.



Object

- Things having physical or logical existence
  - Physical objects
  - Elements of the computer-user environment
  - Data-storage constructs
  - Human entities
  - Collections of data
  - User-defined data types
  - And so on...

Class

- Objects are member of classes.
- Defining the class doesn’t create any objects
- A class is thus a description of a number of similar objects.
  - Himesh, Vishal, and Ankit Tiwary are members of the music composer.
- There is no one person called “music composer,” but specific people with specific names are members of this class if they possess certain characteristics.
- An object is often called an “instance” of a class.

Characteristics of Object-Oriented Languages

- Data abstraction
- Encapsulation
- Inheritance
- Polymorphism
- Reusability

Data abstraction

- Abstraction refers to the act of showing essential features without including its implemetation details.
- It focuses on the outside view(interface) of an object. Therefore, it serves to separate an objects essential behaviour from its implementation.
- It is used to reduce complexity by ignoring some aspect of a subject that are not relevant to the current purpose.
- The type that uses data abstraction are known as Abstract Data Type.

Encapsulation

- The binding of data and functions into a single unit is know as encapsulation.
- Encapsulation achieves information hiding ; hiding all the information that is unnecessary to the outside world
- Each abstraction have two parts : an interface and an implementation.
- Encapsulated elements may be termed as the secrets of abstraction

Inheritance

- It is the process by which objects of one class acquire the properties of objects of another class.
- It supports the concept of **is-a** kind of hierarchical classification
- Each derived class shares common characteristics with the class from which it is derived.
- Provides the idea of reusability

Polymorphism

- “ability to take more than one form”
- An operation may perform different task in different instances.
- The behaviour depends upon the types of data used in the operation.
- Supports function overloading in java
- Using single function name to perform different types of tasks is known as function overloading
- Example: message draw
  - Circle responds with drawing a circle
  - Polygon responds with drawing a polygon
  - …

Benifits of OOP

- Elimination of redundant code via inheritance
- Saves development time
- Data restricted from unauthorized access
- Easily scalable
- Interface between objects is simpler
- Lesser complexity
- Encourages reusability

OOP languages

- Java, C++, C#, Python, PHP, JavaScript, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Common Lisp, MATLAB, Smalltalk and the list goes on...