|  |
| --- |
| **Department of Software Engineering**  **Mehran University of Engineering and Technology, Jamshoro** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Course: SWE121 – Object Oriented Programming** | | | |
| **Instructor** | Mr. Asmatullah Zubair | **Practical/Lab No.** | 05 |
| **Date** | 01-08-2022 | **CLOs** | CLO-3 |
| **Signature** |  | **Assessment Score** | 1 Marks |

|  |  |
| --- | --- |
| **Topic** | **Implementing the concepts of encapsulation** |
| **Objectives** | * What is a class, and how to define that. * How to define and implement the class methods and class constructors. * Understanding the class members and static and non-static contexts. * How to create the object references and how to invoke methods on objects. |

|  |
| --- |
| **Lab Discussion: Theoretical concepts and Procedural steps** |

**Tools:**  Java Development Kit, Text Pad

**Theory**

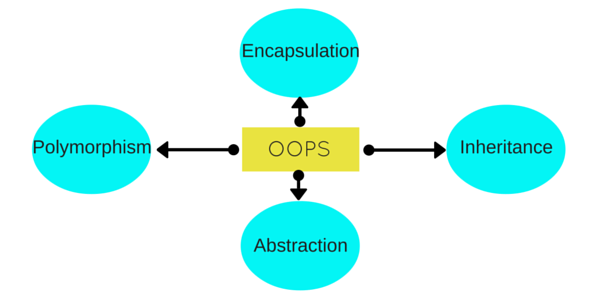
Outline

* Understand the fundamentals of object-oriented programming in Java, including defining classes, class members and create objects instances of a class, invoking methods etc.
* Discuss Data Abstraction and Discuss Data Encapsulation.
* To explain the theory behind constructor and destructor.
* Explaining Access modifiers such private, public and protected.

and their use in a program

# **Object Oriented Programming**

Object Oriented programming is a programming style that is associated with the concept of Class, Objects and various other concepts revolving around these two, like Inheritance, Polymorphism, Abstraction, Encapsulation etc.



Object oriented programming is a way of solving complex problems by breaking them into smaller problems using objects. Before Object Oriented Programming (commonly referred as OOP), programs were written in procedural language, they were nothing but a long list of instructions. On the other hand, the OOP is all about creating objects that can interact with each other, this makes it easier to develop programs in OOP as we can understand the relationship between them.

Object-oriented programming: As the name suggests, Object-Oriented Programming or OOPs refers to languages that uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

OOPs Concepts:

* [Polymorphism](https://www.geeksforgeeks.org/polymorphism-in-java/)
* [Inheritance](https://www.geeksforgeeks.org/inheritance-in-java/)
* [Encapsulation](https://www.geeksforgeeks.org/encapsulation-in-java/)
* [Abstraction](https://www.geeksforgeeks.org/abstraction-in-java-2/)
* [Class](https://www.geeksforgeeks.org/classes-objects-java/)
* [Object](https://www.geeksforgeeks.org/classes-objects-java/)
* [Method](https://www.geeksforgeeks.org/methods-in-java/)
* [Message Passing](https://www.geeksforgeeks.org/message-passing-in-java/)

[](https://media.geeksforgeeks.org/wp-content/cdn-uploads/20190717114649/Object-Oriented-Programming-Concepts.jpg)

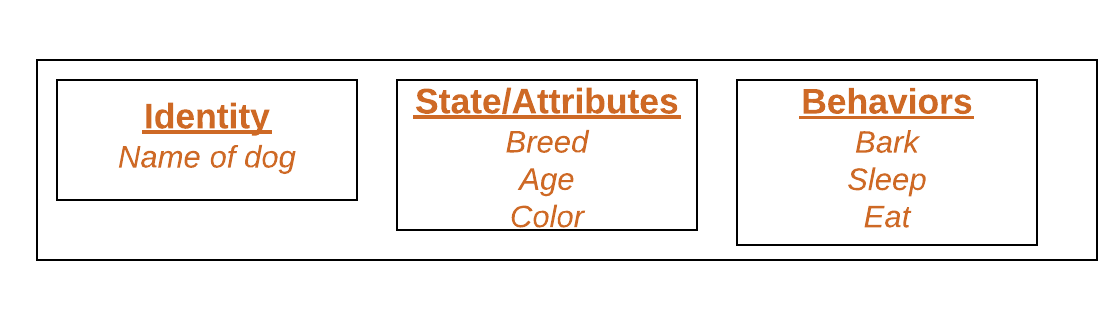
In Object oriented programming we write programs using classes and objects utilizing features of OOPs such as **abstraction**, **encapsulation**, **inheritance** and **polymorphism**

### **Class and Objects**

A class is like a blueprint of data member and functions and object is an instance of class. For example, lets say we have a class **Car** which has data members (variables) such as speed, weight, price and functions such as gearChange(), slowDown(), brake() etc. Now lets say I create a object of this class named FordFigo which uses these data members and functions and give them its own values. Similarly we can create as many objects as we want using the blueprint(class).

1. [Class](https://www.geeksforgeeks.org/classes-objects-java/): A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:
   1. Modifiers: A class can be public or has default access (Refer [this](https://www.geeksforgeeks.org/access-specifiers-for-classes-or-interfaces-in-java/) for details).
   2. Class name: The name should begin with a initial letter (capitalized by convention).
   3. Superclass(if any): The name of the class’s parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
   4. Interfaces(if any): A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
   5. Body: The class body surrounded by braces, { }.
2. [Object](https://www.geeksforgeeks.org/classes-objects-java/): It is a basic unit of Object Oriented Programming and represents the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of:
   1. State : It is represented by attributes of an object. It also reflects the properties of an object.
   2. Behavior : It is represented by methods of an object. It also reflects the response of an object with other objects.
   3. Identity : It gives a unique name to an object and enables one object to interact with other objects.

Example of an object: dog

[](https://media.geeksforgeeks.org/wp-content/uploads/Blank-Diagram-Page-1-5.png)

1. [Method](https://www.geeksforgeeks.org/methods-in-java/): A method is a collection of statements that perform some specific task and return result to the caller. A method can perform some specific task without returning anything. Methods allow us to reuse the code without retyping the code. In Java, every method must be part of some class which is different from languages like C, JAVA and Python.  
   Methods are time savers and help us to reuse the code without retyping the code.

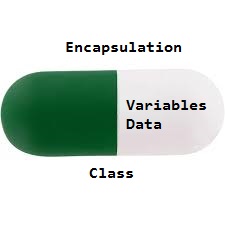
Method Declaration

### **Abstraction**

[Abstraction](https://beginnersbook.com/2017/09/abstraction-in-c-with-example/) is a process of hiding irrelevant details from user. For example, When you send an sms you just type the message, select the contact and click send, the phone shows you that the message has been sent, what actually happens in background when you click send is hidden from you as it is not relevant to you.

### **Encapsulation**

[Encapsulation](https://beginnersbook.com/2017/09/cpp-encapsulation/) is a process of combining data and function into a single unit like capsule. This is to avoid the access of private data members from outside the class. To achieve encapsulation, we make all data members of class private and create public functions, using them we can get the values from these data members or set the value to these data members.

[](https://media.geeksforgeeks.org/wp-content/uploads/Encapsulation.jpg)

### **Inheritance**

[Inheritance](https://beginnersbook.com/2017/08/cpp-inheritance/) is a feature using which an object of child class acquires the properties of parent class. [Inheritance](https://www.geeksforgeeks.org/inheritance-in-java/): Inheritance is an important pillar of OOP(Object Oriented Programming). It is the mechanism in java by which one class is allow to inherit the features(fields and methods) of another class.  
Important terminology:

* + Super Class: The class whose features are inherited is known as superclass(or a base class or a parent class).
  + Sub Class: The class that inherits the other class is known as subclass(or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.
  + Reusability: Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

The keyword used for inheritance is extends.  
Syntax:

class derived-class extends base-class

{

//methods and fields

}

### **Polymorphism**

[Function overloading](https://beginnersbook.com/2017/08/cpp-function-overloading/) and Operator overloading are examples of polymorphism. Polymorphism is a feature using which an object behaves differently in different situation.  
In function overloading we can have more than one function with same name but different numbers, type or sequence of arguments.

[Polymorphism](https://www.geeksforgeeks.org/polymorphism-in-java/): Polymorphism refers to the ability of OOPs programming languages to differentiate between entities with the same name efficiently. This is done by Java with the help of the signature and declaration of these entities.

For example:

|  |
| --- |
| // Java program to demonstrate Polymorphism    // This class will contain  // 3 methods with same name,  // yet the program will  // compile & run successfully  public class Sum {        // Overloaded sum().      // This sum takes two int parameters      public int sum(int x, int y)      {          return (x + y);      }        // Overloaded sum().      // This sum takes three int parameters      public int sum(int x, int y, int z)      {          return (x + y + z);      }        // Overloaded sum().      // This sum takes two double parameters      public double sum(double x, double y)      {          return (x + y);      }        // Driver code      public static void main(String args[])      {          Sum s = new Sum();          System.out.println(s.sum(10, 20));          System.out.println(s.sum(10, 20, 30));          System.out.println(s.sum(10.5, 20.5));      }  } |

**Output:**

30

60

31.0

|  |
| --- |
| **Lab Tasks** |

1. Create a class having 4 functions, add, sub, mul and div. Each function accepts 2 parameters and returns the sum, difference, multiplication and division of these numbers.

Create a main class having main function that uses the above class.

1. Demonstrate the use of the following:

Constructor, Accessors and mutators, static class members.

1. Create a Student class that stores student data, provides methods for initializing and displaying student data. Also provide a parameterized constructor for initializing student class data members.