Website

https://www.curiousinfotechsolutions.com

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Java Programming

Day-5

OOP in Java	https://youtu.be/xXU6GgYKVpQ

Email id:
info@curiousinfotechsolutions.com

Syllabus Day-5

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

What is an OOP?

- ✓Class
- ✓Object
- ✓Method and Message Passing
- ✓Encapsulation
- ✓Abstraction
- ✓Polymorphism
- ✓Inheritance
- ✓Dynamic Binding

What is OOP?What/How

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- OOP stands for Object-Oriented Programming.
- Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions.
- Object-oriented programming has several advantages over procedural programming:
  - OOP is faster and easier to execute.
  - OOP provides a clear structure for the programs.
  - OOP makes the code easier to maintain, modify and debug.
  - OOP makes it possible to create full reusable applications with less code and shorter development time.

Contd..

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- Simula is considered the first object-oriented programming language. The programming paradigm where everything is represented as an object is known as a truly object-oriented programming language.

- Smalltalk is considered the first truly object-oriented programming language.

- The popular object-oriented languages are Java, C#, PHP, Python, C++, etc.

Characteristics of an Object-Oriented Programming languag

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- Following are basic characteristics of OOP
  - Class
  - Object
  - Method and Message Passing
  - Encapsulation
  - Abstraction
  - Polymorphism
  - Inheritance
  - Dynamic Binding

Classes in OOPs

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- The building block of Java that leads to Object-Oriented programming is a Class.
- It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.
- A class is like a blueprint for an object.
- For Example: Consider the Class of Person. There may be many Persons with different names but all of them will share some common properties like all of them will have Name, Age, Height, Gender, Aadhar etc.
- Person is the class and Name, Age, Height, Gender, Aadhar are their properties.

View of Class

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Person

Name,
Age,
Height,
Gender,
Aadhar

Objects in OOPs

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- An Object is an identifiable entity with some characteristics and behavior.
- An Object is an instance of a Class.
- When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.
- When a program is executed, the objects interact by sending messages to one another.
- Each object contains data and code to manipulate the data.
- Objects can interact without having to know details of each other's data or code, it is sufficient to know the type of message accepted and type of response returned by the objects.

Example of Object

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Person

Name,
Age,
Gender

Narendra Modi
Age-71, Male

Amitabh Bachchan
Age-79, Male

Lata Mangeshkar
Age-92, Female

https://www.curiousinfotechsolutions.com

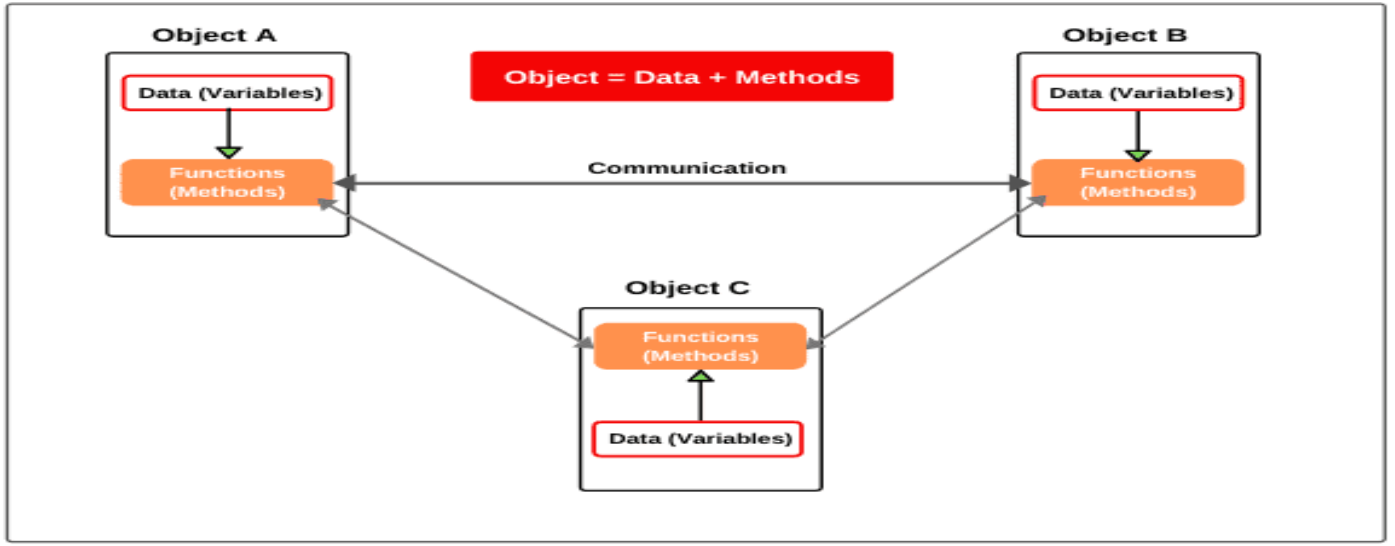
Methods in OOPs

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Data is represented as properties of the object, and behaviors are represented as methods.

For example, a Window object could have methods such **as open and close** , while its state (whether it is open or closed at any given point in time) would be a property.

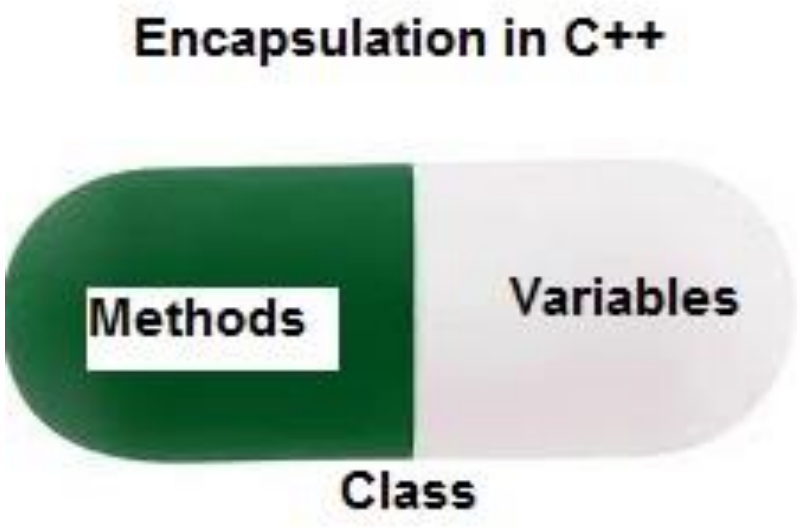
Methods belongs to the class.



Encapsulation in OOPs

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- In normal terms, Encapsulation is defined as wrapping up of data and information under a single unit.
- In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them.



Real-Examples of Encapsulation

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- Consider a real-life example of encapsulation, in a company,
  - there are different sections like the accounts section, finance section, sales section etc.
- The finance section handles all the financial transactions and keeps records of all the data related to finance.
  - Similarly, the sales section handles all the sales-related activities and keeps records of all the sales.
- Now there may arise a situation when for some reason an official from the finance section needs all the data about sales in a particular month.
- In this case, he is not allowed to directly access the data of the sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data.

Abstraction in OOPs

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- Data abstraction is one of the most essential and important feature of object-oriented programming in C++.
- Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.
- Consider a real life example of a man driving a car.
  - *The man only knows that pressing the accelerators will increase the speed of car or applying brakes will stop the car but he does not know about how on pressing accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of accelerator, brakes etc in the car.*

Type of Abstractions

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- *Abstraction in Header files*:
  - One more type of abstraction in C++ can be header files.
  - For example, consider the pow() method present in math.h header file.
  - Whenever we need to calculate the power of a number, we simply call the function pow() present in the math.h header file and pass the numbers as arguments without knowing the underlying algorithm according to which the function is actually calculating the power of numbers.

- *Abstraction using Classes*:
  - We can implement Abstraction in C++ using classes. The class helps us to group data members and member functions using available access specifiers. A Class can decide which data member will be visible to the outside world and which is not.

Contd..

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **Abstraction using access specifiers**

- Access specifiers are the main pillar of implementing abstraction in C++. We can use access specifiers to enforce restrictions on class members. For example:
  - Members declared as **public** in a class, can be accessed from anywhere in the program.
  - Members declared as **private** in a class, can be accessed only from within the class. They are not allowed to be accessed from any part of code outside the class.

Polymorphism in OOPs

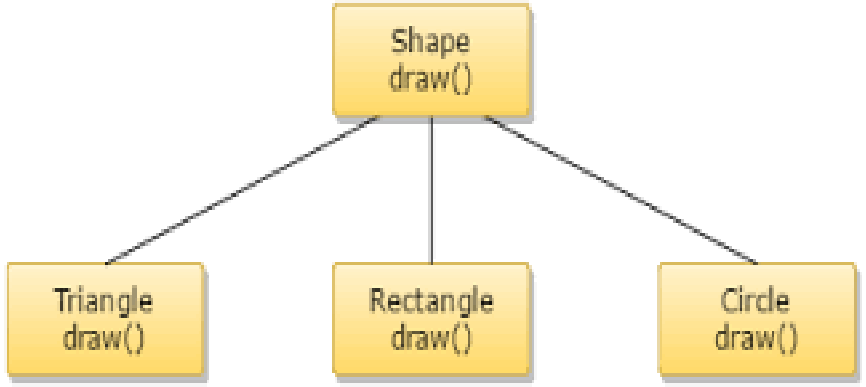
Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.
- A person at the same time can have different characteristic. Like a man at the same time is a father, a husband, an employee. So the same person posses different behavior in different situations. This is called polymorphism.
- An operation may exhibit different behaviors in different instances. The behavior depends upon the types of data used in the operation.

Type of Polymorphism

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- ***Operator Overloading***: The process of making an operator to exhibit different behaviors in different instances is known as operator overloading.
- ***Function Overloading***: Function overloading is using a single function name to perform different types of tasks.



Polymorphism

Inheritance in OOPs

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- The capability of a class to derive properties and characteristics from another class is called Inheritance.
- Inheritance is one of the most important features of Object-Oriented Programming.
  - **Sub Class**: The class that inherits properties from another class is called Sub class or Derived Class.
  - **Super Class**:The class whose properties are inherited by sub class is called Base Class or Super class.
  - **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.

Examples of Inheritance

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Dog, Cat, Cow can be Derived Class of Animal Base Class.