

Encapsulation

Is an OOP technique used to **hide** data from direct access.

USING Encapsulation

Declare **ATTRIBUTES** as **PRIVATE**.

```
modifiers className class{  
  
    //Private Attributes  
  
}  
  
public User class{  
  
    private int userID;  
    private String username;  
    private String firstName, lastname;  
}
```

GETTERS & SETTERS

Are methods used to **get** and **set** encapsulated variables.

SETTER

Is a method used to **set** encapsulated variables.

```
void setUserID(int userID){  
  
    this.userID = userID;  
  
}
```

GETTER

Is a method used to **get** encapsulated variables.

```
Int getUserID(){  
  
    return userID;  
  
}
```

OVERLOADING Constructors

Is an OOP technique used to **create** multiple constructors with different arguments, it is used to cope up with the needs of a certain instance of an object.

PS: This is the same as **Method Overloading**.

```
modifiers class className  
{  
    //Attributes  
    className(arguments)  
    {  
  
    }  
  
    className(arguments)  
    {  
  
    }  
}
```

```
class Employee  
{  
    //Attributes
```

Employee(arguments)

{

}

Employee(arguments)

{

}

}