Sukkur Institute of Business Administration University
Department of Computer Science

Object Oriented Programming

BS – II (CS/SE/AI)
Spring 2025

Lab # 04: To become familiar with Classes and Objects

Instructor: Moona Solangi

Lab Report Rubrics (Add the points in each column, then add across the bottom row to find the total score)					**Total Marks**
S.No	**Criterion**	**0.5**	**0.25**	**0.125**	
1	Accuracy	☐ Desired output	☐ Minor mistakes	☐ Critical mistakes	
2	Timing	☐ Submitted within the given time	☐ 1 day late	☐ More than 1 day late	

Submission Profile

Name:
Enrollment ID:
Comments:

Submission date (dd/mm/yy):
Receiving authority name and signature:

Instructor Signature

Note: Submit this lab hand-out before the next lab with attached solved activities and exercises

Objectives

After performing this lab, students will be able to understand,

- What is Class?
- What is an Object?
- Class vs Object
- Assigning Object References Variables
- Method in class

What is Class?

Class:

A class can be defined as a template/blueprint that describes the behavior/state that the object of its type supports. 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:

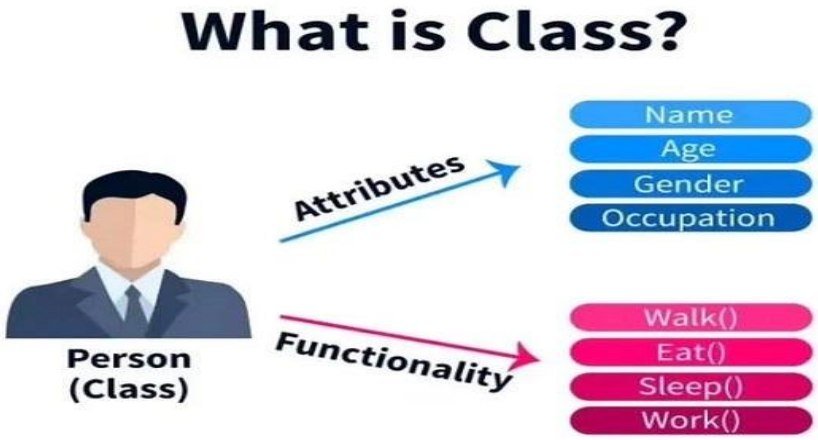
- **Modifiers**: A class can be public or has default access
- **class keyword**: class keyword is used to create a class.
- **Class name**: The name should begin with an initial letter (capitalized by convention).
- **Body**: The class body surrounded by braces, { }.

***Syntax*:**

```
class <class_name>{
    field;
    method;
  }
```

CLASS

A class is a blueprint for declaring and creating objects.



What is an Object?

Objects are the basic units of object-oriented programming.

A simple example of an object would be a person. Logically, you would expect a person to have a name. This would be considered a property of the person. You could also expect a person to be able to do something, such as walking or driving. A typical Java program creates many objects, which as you know, interact by invoking methods.

An object consists of:

- **State**: It is represented by attributes of an object. It also reflects the properties of an object.
- **Behavior**: It is represented by methods of an object. It also reflects the response of an object with other objects.
- **Identity**: It gives a unique name to an object and enables one object to interact with other objects

***Syntax*:**

ClassName ReferenceVariable = new ClassName();

Example:

Car car; **//declare reference to object**

car = new Car(); **//allocate a car object**

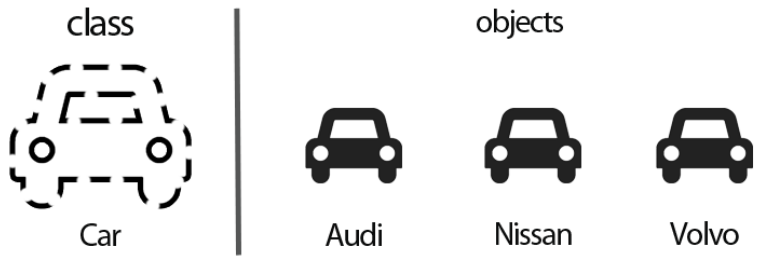
Creating an Object

As mentioned previously, a class provides the blueprints for objects. So basically, an object is created from a class. In Java, the new keyword is used to create new objects.

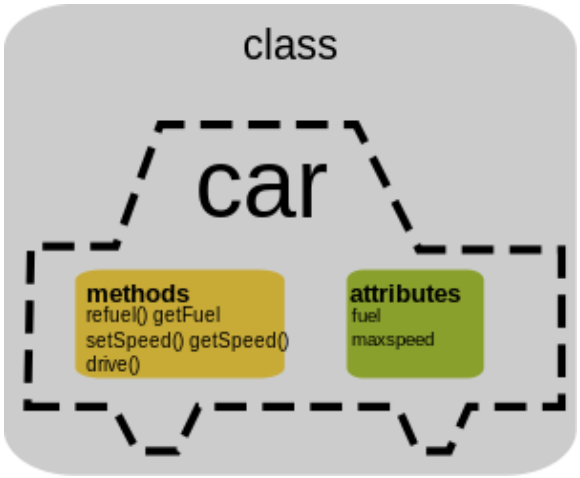
There are three steps when creating an object from a class −

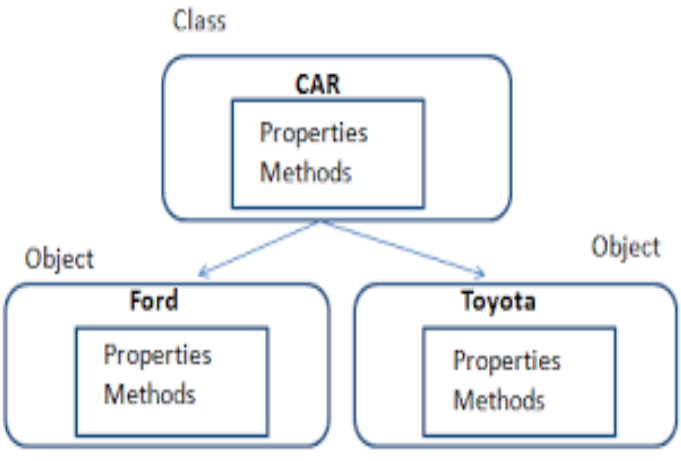
- **Declaration** − A variable declaration with a variable name with an object type.
- **Instantiation** − The 'new' keyword is used to create the object.
- **Initialization** − The 'new' keyword is followed by a call to a constructor. This call initializes the new object.

Class vs Object



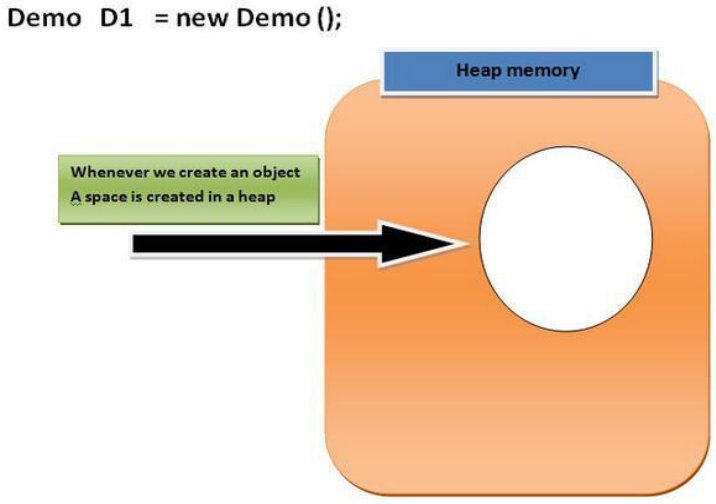
Class containing attributes and methods.



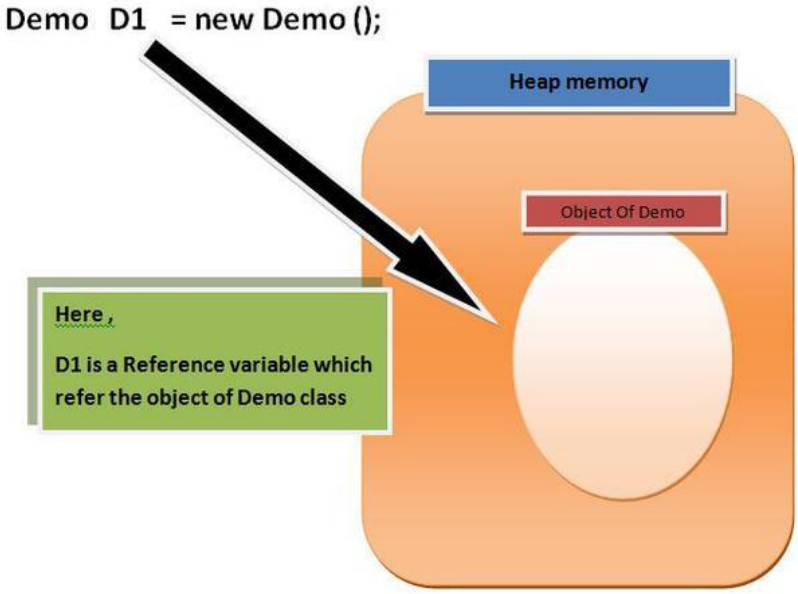


Assigning Object References Variables:

Before We get Started with the Reference variable we should know about the following facts.
When we create an object (instance) of a class then space is reserved in heap memory.



Then, We create a Pointing element or simply called a Reference variable which simply points out the Object (the created space in a Heap Memory).



- Reference variable is used to point object/values.
- Reference variable can also store null value. By default, if no object is passed to a reference variable, then it will store a null value.
- You can access object members using a reference variable using dot syntax.

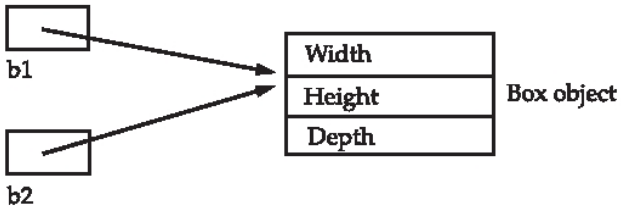
- You can assign value of reference variable to another reference variable.
- Reference Variable is used to store the address of the variable.

Assigning Object Reference Variables does not:

1. Create Distinct Objects.
2. Allocate Memory.
3. Create duplicate Copy.

Consider below example:

Box b1 = new Box();
Box b2 = b1;

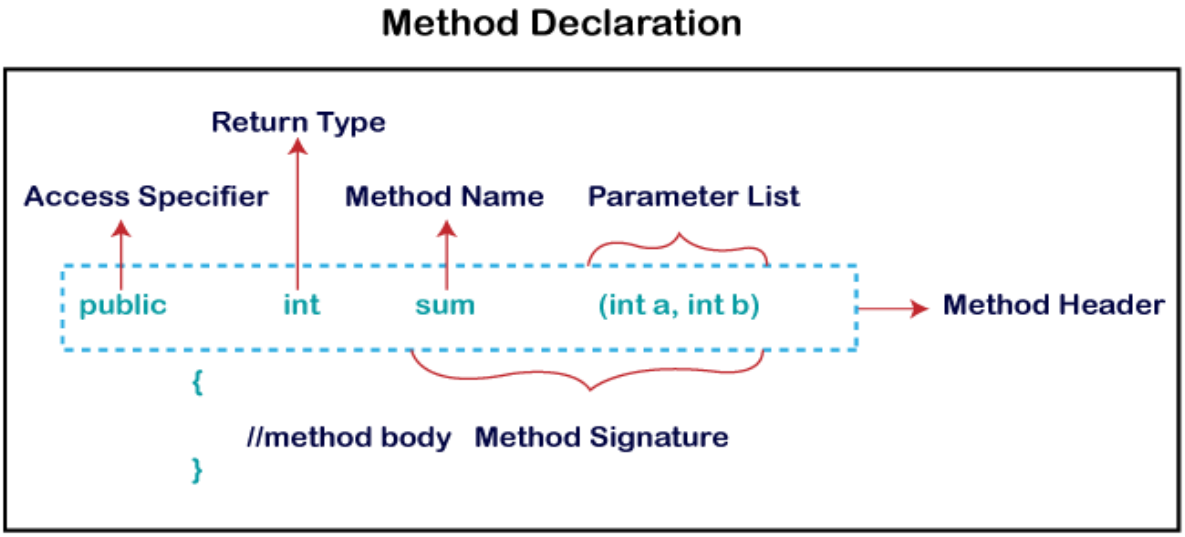


- b1 is reference variable that contains the address of Actual Box Object.
- b2 is another reference variable
- b2 is initialized with b1 means – "b1 and b2" both are referring same object; thus, it does not create duplicate object, nor does it allocate extra memory

Method in class

A method is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation.

- It is used to achieve the reusability of code.
- We write a method once and use it many times.
- We do not require to write code again and again.
- It also provides the easy modification and readability of code, just by adding or removing a chunk of code.
- The method is executed only when we call or invoke it.
- A function is a **combination of instructions** that are combined to achieve some result.



***Syntax*:**

```
returntype name(parameter_list){
  //body of method
}
```

Exercises

Question 1:

Create a class Doctor. The class should have two fields name and qualification. Create setter and getter methods. Using setter method set the name and qualification and using getter method you will display the name and qualification.

Note: Create two setter methods and two getter methods separate for name and qualification.

Question 2:

Create a class named Calculator with four methods:

- **add**(int a, int b): Returns the sum.
- **subtract**(int a, int b): Returns the difference.
- **multiply**(int a, int b): Returns the product.
- **divide**(int a, int b): Returns the quotient.

Create an object of a Calculator and call each method with sample inputs. Display the results of all arithmetic operations.

```
Microsoft Windows [Version 10.0.21996.1]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>javac Main_Cal.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>java Main_Cal
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>
```

Question: 3

Write a program that generate random number from 1-6. It should take input from the user if user enter Y then it should generate another random number. If user enter N then program should display total score.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>javac DiceGame.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>java DiceGame
Welcome to the Dice Game!
You rolled: 3
Do you want to roll again? (Y/N): Y
You rolled: 3
Do you want to roll again? (Y/N): y
You rolled: 1
Do you want to roll again? (Y/N): y
You rolled: 4
Do you want to roll again? (Y/N): n
Your total score is: 11
Thanks for playing!

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>
```

Question: 4

Write a program by creating an 'EmployeeInfo' class having the following methods and print the final salary.

1 - 'getInfo()' which takes the salary, number of hours of work per day of employee as parameter

2 - 'AddWork()' which adds $5 to salary of employee if the number of hours of work per day is more than hours.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>javac EmployeeProgram.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>java EmployeeProgram
Enter employee salary: $200000
Enter number of work hours per day: 7
Final Salary: $200005.0

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>
```

Question: 5

Create a class named **Student** with three fields:

- name (String)
- age (int)
- rollNumber (int)

Create a **method displayInfo()** to print the student's details. Create three **Student** objects and assign different values to each object. Call the displayInfo() method for each object to display student details.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>javac Question4.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>java Question4
Student Name: Tahreem
Age: 20
Roll Number: 101
-------------------------
Student Name: Ahtisham
Age: 21
Roll Number: 102
-------------------------
Student Name: Abdul Ghaffar
Age: 22
Roll Number: 103
-------------------------

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>
```

Question: 6

Write a program that inputs the year a person is born and returns the age of the person.
Example: Person born in 1995 then age is 30.
(import java.time.Year; // To get the current year)

Question: 7

Using Objects in a Real-World Scenario

Description:

1. Create a class **Product** with:

- name (String)
- price (double)

2. Create another class **ShoppingCart** with:
   - An **array of products**.
   - Method addProduct(Product p): Adds a product to the cart.
   - Method calculateTotal(): Returns the total price of all products.
3. In main:
   - Add **3 products** to the cart.
   - Display the **total price**.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>javac SC.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>java SC
Products in Cart:
- Laptop: $800.0
- Mouse: $30.0
- Keyboard: $50.0
Total Cost: $880.0

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>
```

Question: 8

Write a program to print the area of a rectangle by creating a class named 'Area' having two methods. First method named as 'setDim' takes length and breadth of rectangle as parameters and the second method named as 'getArea' returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>javac AreaMain.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>java AreaMain
Enter the length of the rectangle: 4
Enter the breadth of the rectangle: 4
The area of the rectangle is: 16.0

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>
```

Question: 9

Create a Java class representing a Car. The Car class should have attributes such as make, model, year, and color. Implement methods to set and get these attributes. And display using setter and getter methods in CarProgram class.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>javac CarProgram.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>java CarProgram
Car Details:
Make: Toyota
Model: Corolla
Year: 2022
Color: Red

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>
```

Question: 10

Implement a Java class representing a Bank Account. The class should have attributes such as account number, account holder name, and balance. Implement methods to deposit and withdraw money from the account.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>javac BankProgram.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab4_Solutions>java BankProgram
Enter Account Number: 123456789
Enter Account Holder Name: Moona
Enter Initial Balance: $1000

Account Created Successfully!
Account Holder: Moona
Account Number: 123456789
Current Balance: $1000.0

Enter amount to deposit: $
```