Sukkur Institute of Business Administration University
Department of Computer Science

Object Oriented Programming

BS – II (CS/SE/AI)
Spring 2025

Lab # 05: To become familiar with Constructors, new Operator, Parameterized Constructor, and this Keyword.

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,

- **Constructors** initialize objects automatically.
- **The new operator** creates objects dynamically.
- **Parameterized constructors** allow setting values during object creation.
- **this keyword** resolves variable conflicts.

What is a Constructor?

Constructor:

A **constructor in Java** is a block of code, syntactically similar to a method used to initialize an object's state in a class. It has the **same name as the class** and **does not have a return type**.

In other words, a constructor is a special member function of a class used to initialize instance variables of a class. The sole purpose of a constructor is to perform the initialization of data fields of an object in the class.

Syntax to Declare Constructor in Java:

Generally, we declare a constructor inside the public section of the class by the following syntax so that we can create its object in any function. The general syntax to declare a constructor in Java is as follows:

```
JAVA
Access modifiers_name class_name(formal_parameter_list) // constructor header.
{
// Constructor body which is a block of statements.
// Here, we can initialize the values of instance variables.
}
```

The following example code defines a constructor in the class.

```
public class Rectangle
{
  public Rectangle()  {
    // constructor code goes here.
  }
}
```

Here, Rectangle is a class name that must be the same as the name of a class that contains a constructor. That's mandatory.

public is an **access modifier** that indicates that other classes can access the constructor. A constructor can be declared (optionally) as public, protected, and private. These are called access modifiers in Java.

Non-access modifiers cannot be applied with constructors in Java. By mistake, if you apply any other modifiers with the constructor except these three access modifiers, you will get a compile-time error.

Characteristics of Java Constructor:

There are the following characteristics or features of constructor in Java. They are as follows:

1. Constructor's name must be exactly the same as the class name in which it is defined. It must end with a pair of simple braces.

2. The constructor should not have any return type even void also because if there is a return type, JVM would consider as a method, not a constructor.

   Compiler and JVM differentiate constructor and method definitions on the basis of the return type. Suppose you define the method and constructor with the same name as that of the class name, JVM would differentiate between them by using return type. It is a common mistake to declare the void keyword before a constructor. For example:

   ```
   public void Rectangle() {

   }
   ```

   Here, Rectangle() is a method, not a constructor.

3. Java constructor may or may not contain parameters. Parameters are local variables to receive value (data) from outside into a constructor.

4. A constructor is automatically called and executed by JVM at the time of object creation. JVM (Java Virtual Machine) first allocates the memory for variables (objects) and then executes the constructor to initialize instance variables.

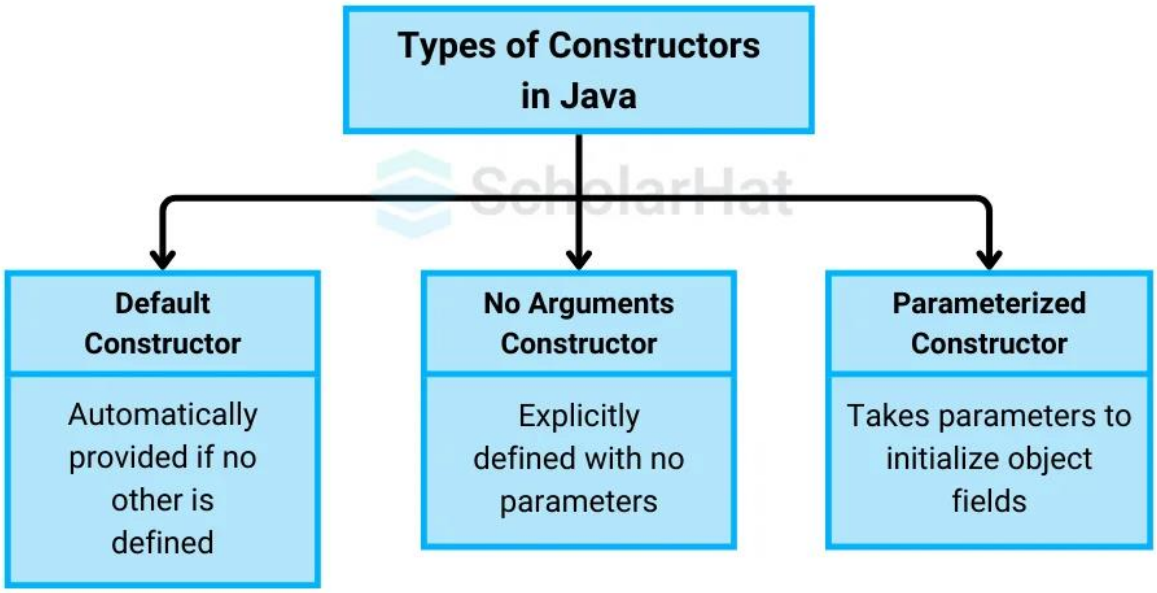
5. It is always called and executed only once per object. This means that when an object of a class is created, constructor is called. When we create second object, the constructor is again called during the second time.

6. Whenever we create an object/instance of a class, the constructor automatically calls by the JVM . If we don't define any constructor inside the class, Java compiler automatically creates a default constructor at compile-time and assigns default values for all variables declared in the class.

   The default values for variables are as follows:

   a. Numeric variables are set to 0.
   b. Strings are set to null.
   c. Boolean variables are set to false.

7. We can do anything in the constructor similar to a method. Using constructors, we can perform all initialization activities of an object.



Example: Default Constructor

```
class Student {

   // Default Constructor

   Student() {

      System.out.println("A new student object is created!");
```

```
    }
}
public class ConstructorExample {
    public static void main(String[] args) {
        Student s1 = new Student(); // Constructor is called automatically
    }
}
```

Output:

A new student object is created!

What is the new operator?

- The **new keyword** is used to create **new objects dynamically** in Java.
- It **allocates memory** and **calls the constructor**.

***Syntax*:**

```
ClassName objectName = new ClassName();
```

Example: Using new to Create Objects

```
class Car {
    Car() {
        System.out.println("A new Car is created!");
    }
}
public class NewOperatorExample {
    public static void main(String[] args) {
        Car car1 = new Car(); // Object creation using `new`
    }
```

```
}
```

Output:

A new Car is created!

What is the “this” keyword in Java?

- **Refers to the current object** of a class.
- **Used to differentiate between instance variables and parameters** when they have the same name.

Example: Using this to Resolve Variable Naming Conflict

```
class Person {
    String name;
    int age;
    // Constructor with `this`
    Person(String name, int age) {
        this.name = name; // Refers to instance variable
        this.age = age;
    }
    void display() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}
public class ThisKeywordExample {
    public static void main(String[] args) {
        Person p1 = new Person("John", 25);
        p1.display();
```

```
    }
}
```

Output:

Name: John, Age: 25

Exercises

Question 1: Create a class Animal that prints "An animal is created!" when an object is instantiated using a default constructor.

Expected Output:

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

C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>javac Q1.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>java Q1
An animal is created!
```

Question 2: Write a program that creates **two objects** of a class Car using the **new** operator.

Question 3: Define a **Student** class with attributes name and grade. Use a **parameterized constructor** to initialize values and display them.

Question 4: Create a class Book with **two constructors**:

- One **default constructor**.
- One **parameterized constructor** to initialize title and author.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>javac Q4.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>java Q4
Book Title: Unknown, Author: Unknown
Book Title: Harry Potter, Author: J.K. Rowling
```

Question 5: Create a class BankAccount with attributes accountNumber, accountHolderName, and balance. Use a **parameterized constructor** to initialize values.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>javac Q5.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>java Q5
Account Number: 123456789
Account Holder: Moona Solangi
Balance: $5000.0
```

Question 6: Create a class Employee with a constructor that calculates a **bonus of 10%** of salary if the experience is **more than 5 years**.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>javac Q6.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>java Q6
Employee Name: Alice
Salary after Bonus: $5500.0
```

Question 7: Create a **BankAccount** class that manages **deposits, withdrawals, and balance inquiries** using **constructors, loops, and scanner input**.

Requirements:

1. Create a BankAccount class with:
    - accountNumber (String)
    - accountHolder (String)
    - balance (double)
2. Implement:
    - **Parameterized Constructor** to initialize account details.
    - **Methods**:
        - deposit(double amount): Adds money to the balance.
        - withdraw(double amount): Deducts money if enough balance is available.
        - displayAccountInfo(): Prints account details.
3. In main:
    - Use Scanner to input account details.
    - Allow the user to **perform multiple transactions** (deposit/withdraw).
    - Use a **loop** to allow repeated transactions until the user **chooses to exit**.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>java Q7
Enter Account Number: 12345
Enter Account Holder Name: Moona Solangi
Enter Initial Balance: $10000

Choose an option:
1. Deposit
2. Withdraw
3. Display Account Info
4. Exit
Enter your choice:
```

Question 8: Develop a **Book** class to manage a **collection of books** using **constructors, loops, and the new operator**.

Requirements:

1. Create a Book class with:
    - title (String)
    - author (String)
    - price (double)
2. Implement:
    - **Parameterized Constructor** to initialize book details.
    - **Method displayBookInfo()** to print book details.
3. In main:
    - Use **Scanner** to allow the user to **add multiple books** (N books).
    - Store books in an **array**.
    - Use a **loop** to display all book details.

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>java Q8
Enter number of books: 3

Enter details for Book 1:
Title: Java Programming
Author: James Gosling
Price: $5000

Enter details for Book 2:
Title: Data Structure
Author: Robert Lafore
Price: $3000

Enter details for Book 3:
Title: Clean Code
Author: Robert Martin
Price: $2000

Library Collection:
1. Title: Java Programming, Author: James Gosling, Price: $5000.0
2. Title: Data Structure , Author: Robert Lafore, Price: $3000.0
3. Title: Clean Code, Author: Robert Martin, Price: $2000.0
```

Question 9: Develop a **Tic-Tac-Toe game** where two players take turns marking **X and O on a 3x3 board.**

Problem Description:

1. The game displays a **3x3 grid.**
2. Two players take turns placing **X** and **O** in a cell.
3. The **game checks for a win** after each move.
    - A player wins if they have **three marks in a row, column, or diagonal.**
    - If all 9 cells are filled and no one wins, it's a **draw.**
4. The program asks **"Do you want to play again?" (Y/N).**

```
C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>javac Q9.java

C:\Users\Moona\Desktop\OOP\Solutions\Lab5_Solutions>java Q9

1 | 2 | 3
---------
4 | 5 | 6
---------
7 | 8 | 9

Player X, enter your move (1-9): 5

1 | 2 | 3
---------
4 | X | 6
---------
7 | 8 | 9

Player O, enter your move (1-9): 8

1 | 2 | 3
---------
4 | X | 6
---------
7 | O | 9
```