Unit – 04: Classes and methods

- Class fundamentals,
- Declaring objects, assigning object to reference variables,
- Constructors (default and parameterized), overloading constructor,
- Understanding this keyword, finalize keyword, static keyword, final keyword,
- garbage collection,
- method introduction and returning a value from a method,
- Overloading method,
- Recursion,
- Introduction to inner and nested classes, command line argument.

Questions to be discussed:

1. What is class in java? How to declare class in java?
2. What is constructor? Explain different types of constructor in java.
3. What do you mean by constructor and destructor?
4. Differentiate between final and finalize in java.
5. What do you mean by method in java? Why we use method in java program?
6. Discuss about method overloading and constructor overloading in java.
7. What do you mean by recursive method? Explain it with example.
8. Write short notes on:
    a. This keyword
    b. Static keyword
    c. Garbage collection

What is an object in Java?

- An object is a real-world entity.
- An entity that has state and behavior is known as an object.
- Example: chair, bike, marker, pen, table, car, etc.
- An object is an instance of a class.
- A class is a template or blueprint from which objects are created.
- So, an object is the instance (result) of a class.

How to Create Object in Java?

- The **object** is a basic building block of an OOPs language.
- In **Java**, we cannot execute any program without creating an **object**.
- Using the **new** keyword we create an object of the class.

  Syntax:

```
ClassName object = new ClassName( );
```

Assigning object to reference variables:

- When we create an object of class then space is reserved in heap memory.
- Now, the space in the memory is created but how to access that space.
- Then, we create a reference variable which simply points out the Object.
- So simply we can say that the reference variable is used to point an object.
- By default, if no object is passed to a reference variable then it will store a null value.

  Example:

```
Myclass D1= new Myclass( );
int i = 10;
```

What is a class in Java?

- A class is a group of objects which have common properties.
- It is a template or blueprint from which objects are created.
- It is a logical entity, it can't be physical.
- A class in Java can contain property and methods.

How to declare class in java?

- Java provides a reserved keyword **class** to define a class.
- The keyword must be followed by the class name.
- Inside the class, we declare methods and variables.

```
class class_name
{
    // member variables
    // class methods
}
```

Difference between object and class:

Object	Class
Object is an instance of a class.	It is a blueprint from which objects are created.
Object is a real world entity such as pen, laptop, mobile, bed, mouse, chair etc.	Class is a group of similar objects.
Object is a physical entity.	Class is a logical entity.
Object is created through new keyword.	Class is declared using class keyword.
Object is created many times as per requirement.	Class is declared once.
Object allocates memory when it is created.	Class doesn't allocated memory when it is created.

Constructor in java:

- Constructor are the special member function.
- It is used to initialize the data members of the class.
- Constructor has the same name as the class name.
- A constructor has no return type not even void.
- Constructors are automatically invoked (call) as soon as the object of its class is created.
- It is called constructor because it constructs the values at the time of object creation.
- It is not necessary to write a constructor for a class, because java compiler creates a default constructor if your class doesn't have any.

How to declare Constructor?

- As we know that constructor has the same name as the class, thus it can be easily identified.
- Syntax:

```
class_name (arguments if any)
{
  ...
  ...
};
```

Types of constructor:

- There are two types of constructors in Java:
  1. Default constructor, and
  2. Parameterized constructor.

Default Constructor:

- A constructor having no arguments is called "Default Constructor.
- The default constructor is used to provide the default values to the object like 0, null, etc.

Parameterized Constructor:

- A constructor having arguments is called a parameterized constructor.
- The parameterized constructor is used to provide different values to distinct objects.
- We can have any number of parameters in the constructor.

Constructor overloading in Java:

- In Java, we can overload constructors like methods.
- It can be defined as the concept of having more than one constructor with different parameters so that every constructor can perform a different task.
- They are differentiated by the compiler by the number of parameters in the list and their types.

  Example:

```
class Student
{
        variables of the class
        Student()
        {
                ...........
        }
        Student(parameters)
        {
                ...........
        } ;
}
```

Rules for creating Java constructor:

1. Constructor name must be the same as its class name.
2. A Constructor must have no return type.
3. A Java constructor cannot be abstract, static, final, and synchronized.

What is Destructor?

- A destructor is a member function of a class that deallocates the memory allocated to an object.
- A destructor is also declared and defined with the same name as that of the class.
- A destructor is preceded by a **tilde (~)** symbol.
- A single class has only a single destructor.

How to declare destructor?

Syntax:

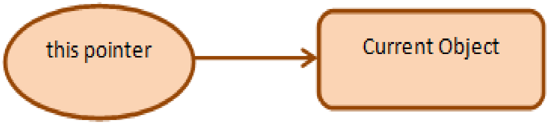
```
~ class_name (no arguments)
{
  …
  …
};
```

Difference between constructor and destructor:

CONSTRUCTOR	DESTRUCTOR
It allocates the memory to an object.	It deallocates the memory of an object.
class_name(arguments if any){ };	~ class_name(no arguments){ };
Constructor accepts argument	Destructor does not accept any argument.
Constructor is called automatically, while the object is created.	Destructor is called automatically, as block of program terminates.
There can be multiple constructors in a class.	There is always a single destructor in the class.
Constructors can be overloaded.	Destructor can not be overloaded.

This keyword:

- It is a keyword that refers to the current object in a method or constructor.
- If the name of instance variable and local variable are same then at the runtime JVM gets confused.
- To avoid this type of problem "this" keyword is used.
- It is used to call default constructor of its own class.
- It is used to called parametrized constructor of its own class.
- "this" can be passed as argument in the constructor call.
- "this" can be used to return the current class instance from the method.



Static keyword:

- It is a keyword which is used to make a variable or method constant.
- If you declare any variable as static, it is known as a static variable.
- The static keyword is used for efficient memory management.
- The static variable gets memory only once in the class area at the time of class loading.
- We can apply static keyword with variables, methods, blocks and nested classes.

Final Keyword:

- The final keyword in java is used to restrict the user.
- Using the final keyword means that the value can't be modified in future.
- The **final** keyword can be used with class method and variable.
- A final class cannot be inherited, a final method cannot be overridden and a final variable cannot be reassigned.

Difference between final, finally and finalize:

Final	finally	finalize
final is the keyword which is used to apply restrictions on a class, method or variable.	finally is the block in Java Exception Handling to execute the important code whether the exception occurs or not.	finalize is the method in Java which is used to perform clean up processing just before object is garbage collected.
Final keyword is used with the classes, methods and variables.	Finally block is always related to the try and catch block in exception handling.	finalize() method is used with the objects.
Once declared, final variable then it cannot be modified, overridden & inherited.	finally block runs the important code even if exception occurs or not & cleans up all the resources used in try block	Finalize method performs the cleaning activities with respect to the object before its destruction.
Final method is executed only when we call it.	Finally block is executed as soon as the try-catch block is executed.	finalize method is executed just before the object is destroyed.

What are Methods in Java?

- A method in Java is a block of code that performs specific actions mentioned in it.
- Method runs only when it is called.
- You can insert values or parameters into methods, and they will only be executed when called.
- They are also referred to as functions.

Why we use methods in Java?

- It allows code reusability (define once and use multiple times)
- You can break a complex program into smaller chunks of code
- It increases code readability

How to Declare Methods in Java?

- You can only create a method within a class.

Syntax:

```
Return_type function_name (parameters)
 {
          //method body
 }
```

Return Type:

- It defines the return type of the method.
- In the above syntax, “int” is the return type.
- We can mention void as the return type if the method returns no value.

Method Overloading in Java:

- Method Overloading is a feature that allows a class to have multiple methods with the same name but with different number, sequence or type of parameters.
- In short multiple methods with same name but with different signatures**.**
- For example:

```
add(int a, int b)          //having two int parameters
add(int a, int b, int c)          //having three int parameters
```

- Method overloading increases the readability of the program.
- There are different ways to overload the method:
  1. By changing number of arguments
  2. By changing the data type of parameters
  3. Changing the order sequence of parameters

Recursion in Java:

- The process in which a method calls itself directly or indirectly is called recursion.
- A method in java that calls itself is called recursive method.
- It makes the code compact but complex to understand.
- Using recursive algorithm, certain problems can be solved quite easily.

factorial(n) = n * factorial(n - 1) *..........1   where n ≥ 1

Exp: factorial(5) = 5 * 4 * 3 * 2* 1 = 120

Syntax:

```
return_type method_name( )
{
        //code to be executed
        method_name();//calling same method
}
```

Write a Java program to find the factorial of a number?

```
public class Main
{
        static int factorial(int n)
        {
                if(n==0 || n=1)
                {
                        return 1;
                }
                else
                {
                        return n * factorial(n-1);
                }
        }
        public static void main(String[] args)
        {
                Int x = 5;
                System.out.println("The value of factorial x is:" + factorial(x));
        }
}
```

Garbage collection:

- When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program.
- Eventually, some objects will no longer be needed.
- The garbage collector finds these unused objects and deletes them to free up memory.
- It is the process by which Java programs perform automatic memory management.
- In other words, it is a way to destroy the unused objects.

Java Inner Classes (Nested Classes):

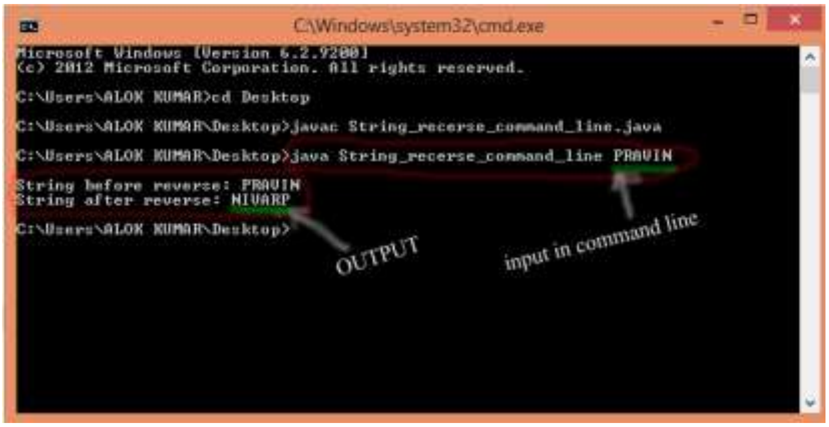
- In java, just like data member and member function we can also create a class as its member.
- The class is written within the class is called inner class.
- The class that holds the inner class is called outer class.
- The inner class is also known as nested class.

  Example:

```
Class outer
{
        int a;
        void add()
        {       }
        class inner
        {
        }
}
```

Command-line argument:

- It is an argument i.e. passed at the time of running the Java program.
- In the command line, the arguments passed from the console can be received in the java program and they can be used as input.
- The users can pass the arguments during the execution bypassing the command-line arguments inside the main( ) method.
- We need to pass the arguments as space-separated values.
- We can pass both strings and primitive data types (int, double, float, char, etc) as command-line arguments.



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\ALOK KUMAR>cd Desktop

C:\Users\ALOK KUMAR\Desktop>javac String_recerse_command_line.java

C:\Users\ALOK KUMAR\Desktop>java String_recerse_command_line PRAVIN

String before reverse: PRAVIN
String after reverse: NIVARP

C:\Users\ALOK KUMAR\Desktop>
```