

Learning Java - A Foundational Journey

Session: 1

Introduction to Java





- ◆ Explain the structured programming paradigm
- ◆ Explain the object-oriented programming paradigm
- ◆ Explain the features of Java as a OOP language
- ◆ Describe Java platform and its components
- ◆ List the different editions of Java
- ◆ Explain the evolution of Java Standard Edition (Java SE)
- ◆ Describe the steps for downloading and installing Java Development Kit (JDK)

For Aptech Centre Use Only



- ◆ The most prominent use of computers is to solve problems quickly and accurately.
- ◆ The solution adopted to solve a problem is provided as a sequence of instructions or specifications of activity which enables a user to achieve the desired result.



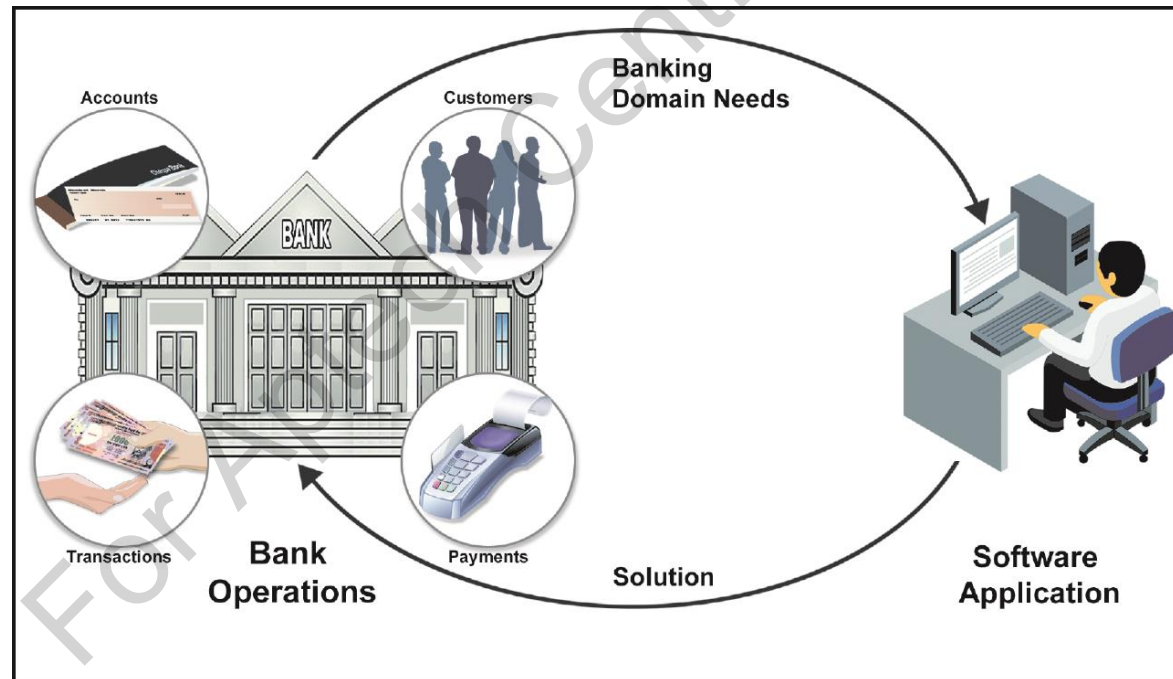
Software Applications

- The solution for solving a problem in the field of information technology is achieved by developing software applications.
- A software application can be defined as a collection of programs that are written in high-level programming languages to solve a particular problem.



◆ Knowledge of Domain:

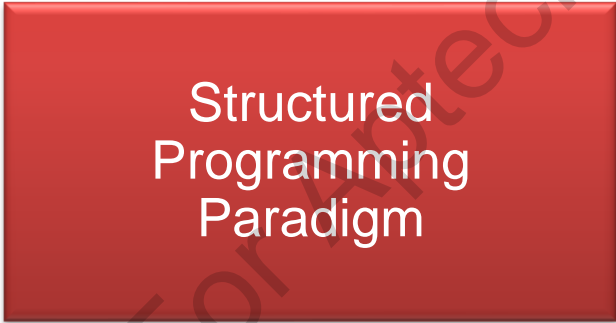
- ◆ Plays an important role while developing software applications
- ◆ Can be defined as field of business or technology to which a problem belongs
- ◆ Figure shows development of software application as a solution for managing various operations in a banking domain.





◆ Programming Languages:

- ◆ Development of software application is done using a programming language.
- ◆ A programming language is used as a medium for communicating instructions to the computer.
- ◆ Programming language enforces a particular style of programming that is referred to as a programming paradigm.
- ◆ Two types of programming paradigm:

A red rectangular box with a slight gradient and shadow, containing the text "Structured Programming Paradigm" in white.

Structured
Programming
Paradigm

A blue rectangular box with a slight gradient and shadow, containing the text "Object-oriented Programming Paradigm" in white.

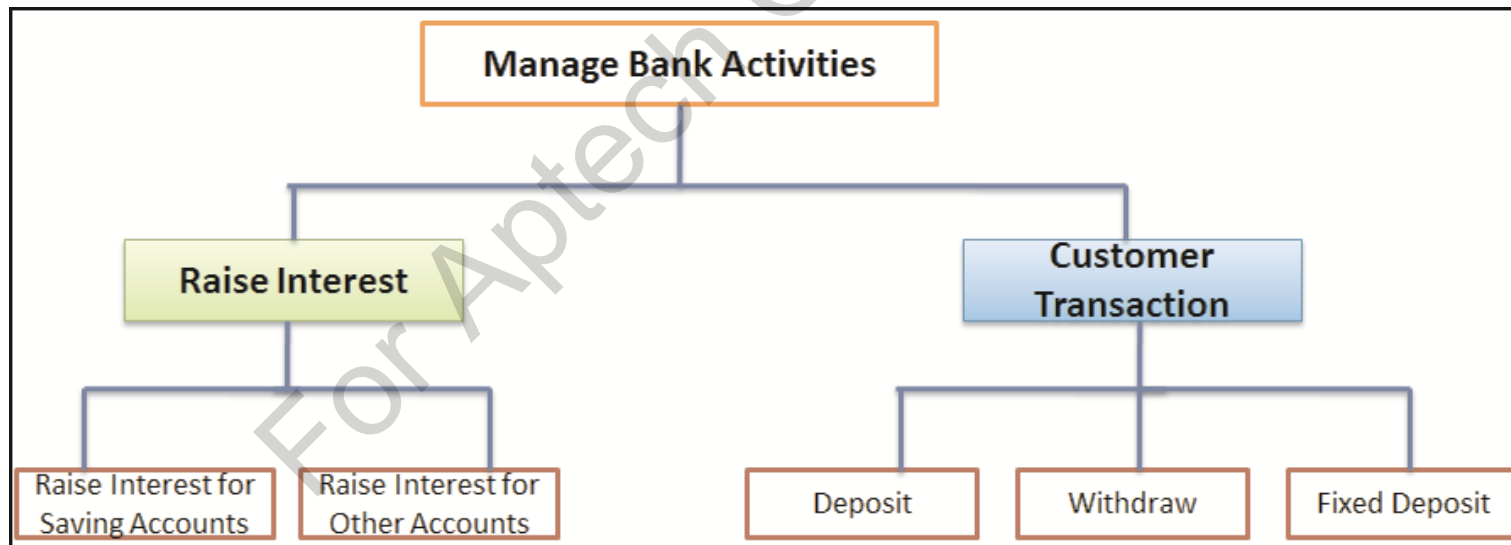
Object-oriented
Programming
Paradigm



Structured Programming

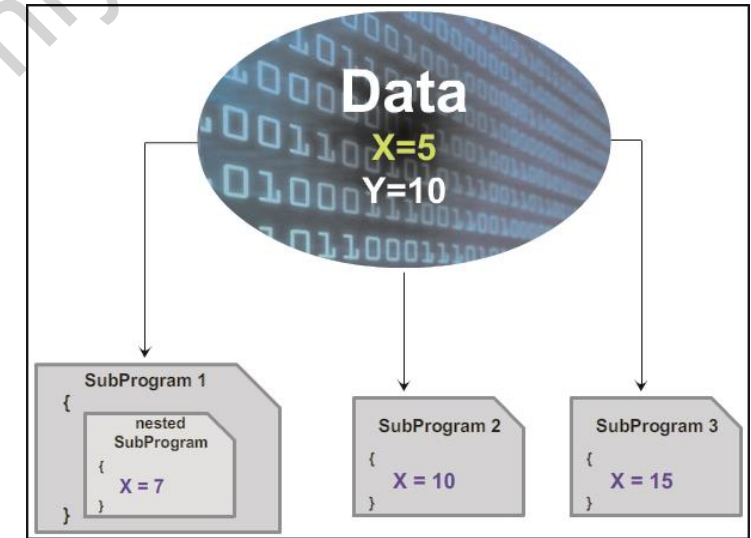
- In structured programming paradigm, the application development is decomposed into a hierarchy of subprograms.
- The subprograms are referred to as procedures, functions, or modules in different structured programming languages.
- Each subprogram is defined to perform a specific task.
- Some of structured programming languages are C, Pascal, and Cobol.

◆ Figure displays bank application activities broken down into subprograms:





- ◆ Main disadvantage of structured programming languages are as follows:
 - ◆ Data is shared globally between the subprograms.
 - ◆ Efforts are spent on accomplishing the solution rather than focusing on problem domain.
- ◆ This often led to a software crisis, as the maintenance cost of complex applications became high and availability of reliable software was reduced.

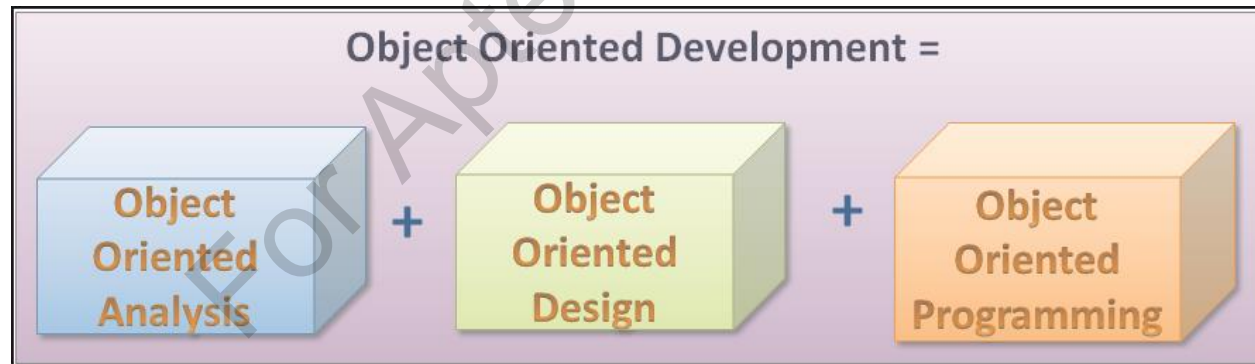


Object-oriented Programming Paradigm 1-3



- ◆ Growing complexity of software required change in programming style.
- ◆ Some features that were aimed are:
 - ◆ Development of reliable software at reduced cost.
 - ◆ Reduction in the maintenance cost.
 - ◆ Development of reusable software components.
 - ◆ Completion of software development with the specified time interval.
- ◆ These features resulted in the evolution of **object-oriented programming paradigm**.
- ◆ Software applications developed using object-oriented programming paradigm are:
 - ◆ Designed around data, rather than focusing only on functionalities.
- ◆ Figure shows different activities involved in object-oriented software development:

CHECKLIST





Object-oriented Analysis (OOA) phase determines the functionality of the system.

Object-oriented Design (OOD) phases determines the process of planning a system in which objects interact with each other to solve a software problem.

Object-oriented Programming (OOP) deals with the actual implementation of the application.

- ◆ Unified Modeling Language (UML) helps to create visual models in the system.
- ◆ The actual implementation of these visual models is done using an OOP language.

Object-oriented Programming Paradigm 3-3



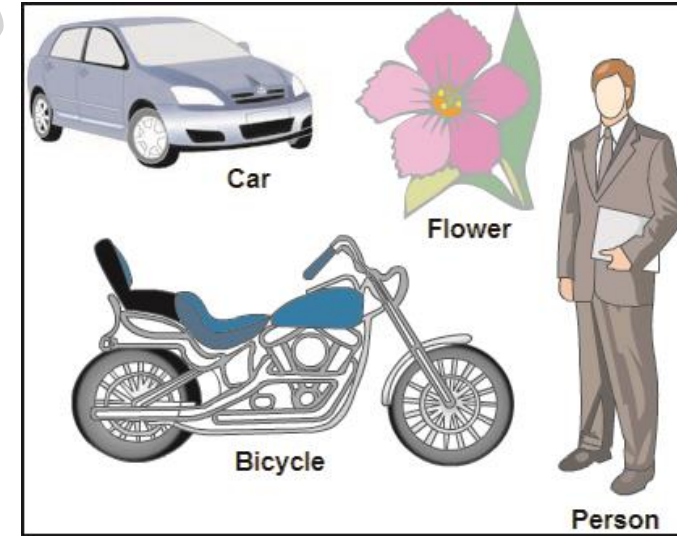
- ◆ An OOP language is based on certain principles that are as follows:

Object	Represents an entity which possesses certain features and behaviors.
Class	Is a template that is used to create objects of that class.
Abstraction	Is a design technique that focuses only on the essential features of an entity for a specific problem domain.
Encapsulation	Is a mechanism that combines data and implementation details into a single unit called class.
Inheritance	Enables the developer to extend and reuse the features of existing classes and create new classes. The new classes are referred to as derived classes.
Polymorphism	Is the ability of an object to respond to same message in different ways.

Concept of an Object 1-2



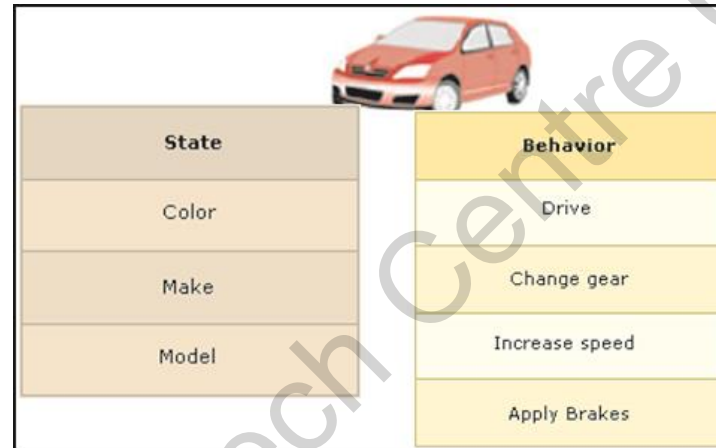
- ◆ An object represents a real-world entity.
- ◆ Any tangible or touchable entity in the real-world can be described as an object.
- ◆ Figure shows some real-world entities:
- ◆ Each object has:
 - ◆ **Characteristics** – Defined as attributes, properties, or features describing the object.
 - ◆ **Actions** – Defined as activities or operations performed by the object.
- ◆ Example of an object, **Dog**.
 - ◆ **Properties** – Breed, Color, and Age
 - ◆ **Actions** – Barking, Eating, and Running
- ◆ The concept of objects in the real-world can be extended to the programming world where software 'objects' can be defined.



Concept of an Object 2-2



- ◆ A software object has state and behavior.
- ◆ **'State'** refers to object's characteristics or attributes.
- ◆ **'Behavior'** of the software object comprises its actions.
- ◆ Following figure shows a software object, a **Car** with its state and behavior:

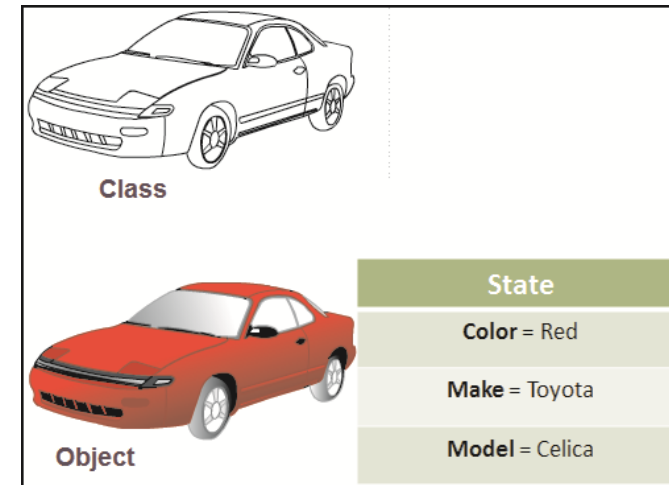


- ◆ Advantages of using objects are as follows:

- 1 • They help to understand the real-world.
- 2 • They map attributes and actions of real-world entities with state and behavior of software objects.

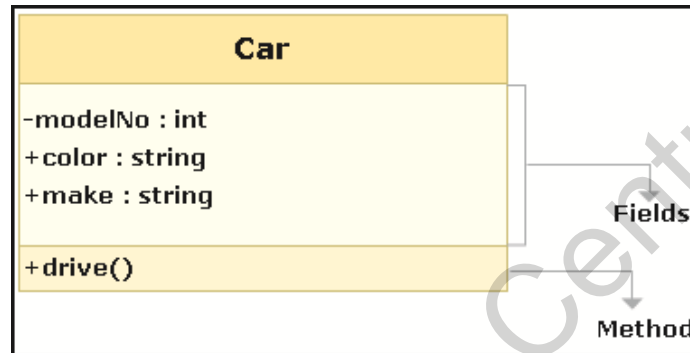


- ◆ In the real-world, several objects:
 - ◆ Have common state and behavior.
 - ◆ Can be grouped under a single class.
 - ◆ Example: All car objects have attributes, such as color, make, or model.
- ◆ **Class:**
 - ◆ Can be defined that a class is a template or blueprint which defines the state and behavior for all objects belonging to that class.
 - ◆ Following figure shows a car as a template and a Toyota car as an object or instance of the class:





- ◆ Class comprises fields and methods, collectively called as members.
 - ◆ **Fields** – Are variables that depict the state of objects.
 - ◆ **Methods** – Are functions that depict behavior of objects.

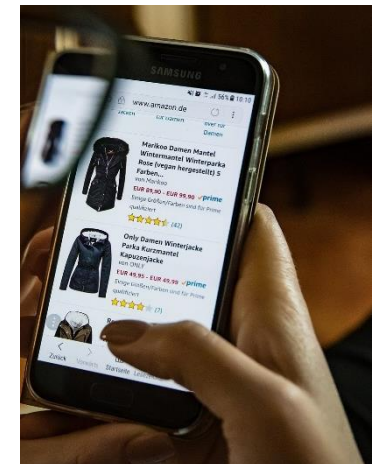


- ◆ Following table shows the difference between a class and an object:

Class	Object
Class is a conceptual model	Object is a real thing
Class describes an entity	Object is the actual entity
Class consists of fields (data members) and functions	Object is an instance of a class



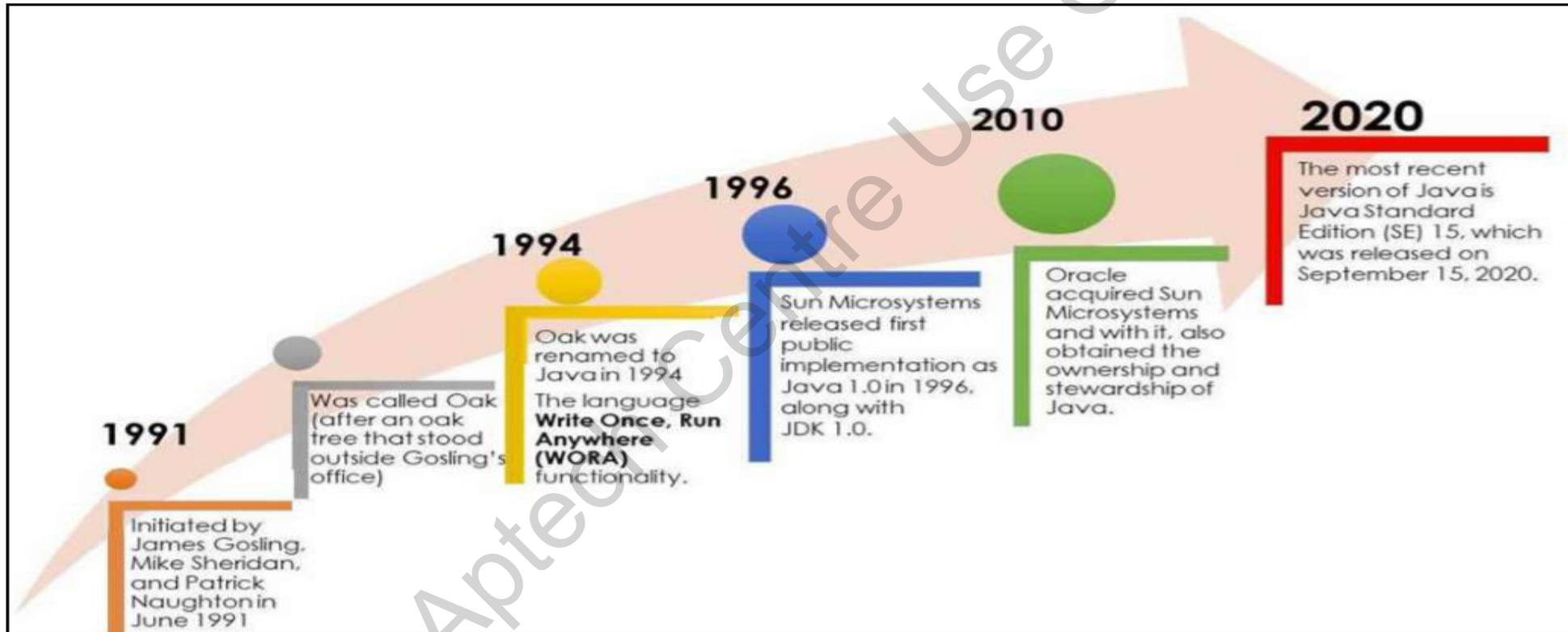
- ◆ It is one of the most popular OOP language.
- ◆ It helps programmers to develop wide range of applications that can run on various hardware and Operating System (OS).
- ◆ It is also a platform that creates an environment for executing Java application.
- ◆ It caters to small-scale to large-scale problems across the Internet.
- ◆ Java applications are built on variety of platforms that range from:
 - ◆ Embedded devices to desktop applications
 - ◆ Web applications to mobile phones
 - ◆ Large business applications to supercomputers



History of Java



- ◆ The brief timeline of Java is depicted as follows:



Timeline of Java

Since James Gosling was the key member of the team that developed the language, he is known as the father of Java.



Java Virtual Machine (JVM)

- Programming languages, such as C and C++, translate compiled code into an executable binary code, which is machine dependent. JVM is an execution engine that creates a platform independent execution environment for executing Java compiled code.

Java Runtime Environment (JRE)

- JRE is also written as Java RTE. The JRE is a set of software tools that are used for developing Java applications. It is used to provide runtime environment. It is the implementation of JVM.

Java Development Kit (JDK)

- JDK is a binary software development kit released by Oracle Corporation as part of the Java platform. It is an implementation of Java and distributed for various platforms, such as Windows, Linux, Mac OS X, and so on. JDK contains a private JVM and a few other resources such as an interpreter, a compiler (javac), an archiver (jar), a documentation generator (Javadoc), and so on.

Differences between Java SE, JRE, and JDK



	JRE	Java SE
Who requires it?	Computer users who run applications written using Java technology	Software developers who write applications using Java technology
What is it?	An environment required to run applications written using Java programming language	A software development kit used to write applications using Java programming language
How do you get it?	Distributed freely and is available from: <i>java.com</i>	Distributed freely and is available from: <i>oracle.com/javase</i>

JRE	JDK
Implementation of the JVM which actually executes Java programs.	A bundle of software that you can use to develop Java based applications.
JRE is a plug-in required for running Java programs.	Java Development Kit is required for developing Java applications.
JRE is smaller than the JDK so it requires less disk space.	JDK requires more disk space as it contains JRE along with various development tools.
JRE can be downloaded/supported freely from: <i>java.com</i>	JDK can be downloaded/supported freely from <i>oracle.com/technetwork/java/javase/downloads/</i>
It includes the JVM, Core libraries, and other additional components to run applications written in Java.	It includes the JRE, set of API classes, Java compiler, Webstart, and additional files required to write Java applications.



- ◆ OpenJDK refers to a free and open-source implementation of Java.
- ◆ Oracle JDK has had better performance than OpenJDK.
- ◆ However, the performance of OpenJDK is growing.
- ◆ Contributions of the OpenJDK community often outperform Oracle JDK.
- ◆ Both OpenJDK and Oracle JDK are created and maintained currently by Oracle only.

For Aptech Centre Use Only



- ◆ Class Data Sharing (CDS) feature helps reduce the startup time and memory footprint between multiple JVMs.
- ◆ When you use the installer to install the Oracle JRE, the installer loads a default set of classes from the system Java Archive (JAR) file into a private internal representation.
- ◆ The memory for applications comprises two components namely, stack and heap. The stack is an area in the memory that stores object references and method information. The heap area of memory deals with dynamic memory allocations.
- ◆ A portion of the shared archive on the same host is mapped as read-only and shared among multiple JVM processes.
- ◆ Application Class-Data Sharing
 - ◆ Allows application classes to be placed in a shared drive.
 - ◆ The common class metadata is shared across different Java processes.
 - ◆ AppCDS allows the built-in system class loader, built-in platform class loader, and custom class loaders to load the archived classes.

Downloading and Installing the JDK



- ◆ To download the JDK, follow the link <https://www.oracle.com/java/technologies/javase-jdk15downloads.html>.

Product / File Description	File Size	Download
Linux ARM 64 RPM Package	141.82 MB	jdk-15.0.2_linux-aarch64_bin.rpm
Linux ARM 64 Compressed Archive	157 MB	jdk-15.0.2_linux-aarch64_bin.tar.gz
Linux x64 Debian Package	154.81 MB	jdk-15.0.2_linux-x64_bin.deb
Linux x64 RPM Package	162.03 MB	jdk-15.0.2_linux-x64_bin.rpm
Linux x64 Compressed Archive	179.35 MB	jdk-15.0.2_linux-x64_bin.tar.gz
macOS Installer	175.93 MB	jdk-15.0.2_osx-x64_bin.dmg
macOS Compressed Archive	176.51 MB	jdk-15.0.2_osx-x64_bin.tar.gz

You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software. X

☒ I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE

Download jdk-15.0.2_windows-x64_bin.exe

For Aptech Centre Use Only

Setting the PATH Environment Variable



- ◆ To set the PATH variable permanently, add the full path of the jdk-15\bin directory to the PATH variable.
- ◆ Typically, the full path is:
C:\Program Files\Java\jdk-15\bin To set the PATH variable on Microsoft Windows:
- ◆ Select Control Panel and then, System.
- ◆ Click Advanced and then, Environment Variables.
- ◆ Add the location of the bin folder of the JDK installation to the PATH variable in System Variables.

For Aptech Centre Use Only



- ◆ **What is Project Graal?**

Project Graal was created by Oracle with the ultimate goal of improving JVM based languages performance to match performance levels that native languages enjoy.

- ◆ **Just-In-Time Compiler** - Is a form of dynamic compilation that combines the use of Ahead-Of-Time (AOT) compilation and interpretation, and thus, has advantages and disadvantages from both technologies.
- ◆ **Ahead-Of-Time Compilation** - Main goal was to improve the start-up time of Java applications with minimum impact to the application's peak performance.
- ◆ **Interpretation** - It is a process that interprets code and performs specified actions. There are different ways an interpreter can work: it can parse the code and execute it directly, turn the code into an intermediate representation and execute that, or it can execute code that has already been pre-compiled.

For Aptechn Centre Use Only

New Features of Java



Java Version	Features
Java 9	Private and Static methods in interfaces
	Try with Resources Improvements
	Immutable Collections
	Optional Class Improvements
	Enhanced @Deprecated annotation
	Stream API Improvements
	JShell
	Create and Use Modules
Java 10	Time-Based Release Versioning
	Local-Variable Type Inference
	Experimental Java-Based JIT Compiler
	Application Class-Data Sharing
	Parallel Full GC for G1
	Garbage-Collector Interface
	Additional Unicode Language-Tag Extensions
	Root Certificates
	Thread-Local Handshakes
	Heap Allocation on Alternative Memory Devices
	Remove the Native-Header Generation Tool – javah
	Consolidate the JDK Forest into a Single Repository
Java 11	Running Java File with single command
	New utility methods in String class
	Local-Variable Syntax for Lambda Parameters
	Nested Based Access Control
	Flight Recorder
	Reading/Writing Strings to and from the Files
Java 12	String API Updates
	Compact Number Formatting
	File mismatch() Method
	Teeing Collectors in Stream API
Java 13	Reimplement the Legacy Socket API
	Dynamic CDS Archive
	FileSystems.newFileSystem() Method
	Support for Unicode
	DOM and SAX Factories with Namespace
Java 14	Switch Expressions
	Pattern Matching for instanceof
	Helpful NullPointerExceptions
	Records
	Text Blocks
Java 15	Hidden classes
	Z Garbage Collector
	Sealed classes
	Records
	Improved security with Edwards-Curve Digital Signature algorithm

Structure of a Java Class



package

- A package defines a namespace that stores classes with similar functionalities in them.

import

- The import keyword identifies the classes and packages that are used in a Java class.

class

- The class keyword identifies a Java class. It precedes the name of the class in the declaration.

Variables

- They represent the state of objects.

Methods

- Functions that represent some action to be performed on an object.

Constructors

- Methods or functions that are invoked during the creation of an object.

```
package <package_name>;

import <other_packages>;

public class ClassName {
    <variables(also known as fields)>;

    <constructor method(s)>;

    <other methods>;
}
```



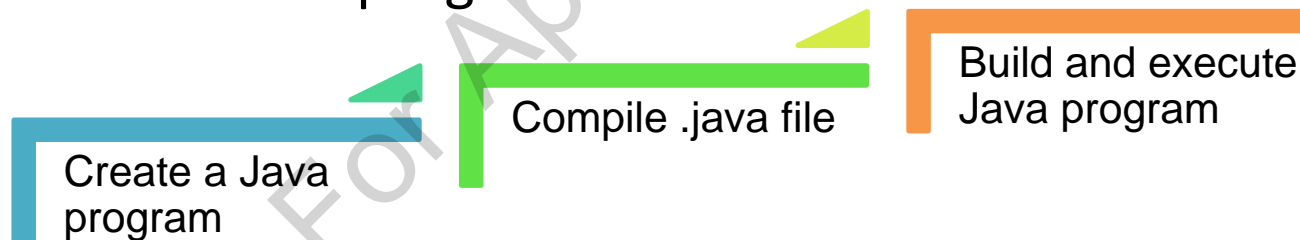
◆ Basic requirements to write a Java program are as follows:

1. JDK 15 installed and configured on the system
2. A text editor

The text editor can be any simple editor included with the platforms. For example, the Windows platform provides a simple text editor named Notepad.

To create, compile, and execute a Java program, perform following steps:

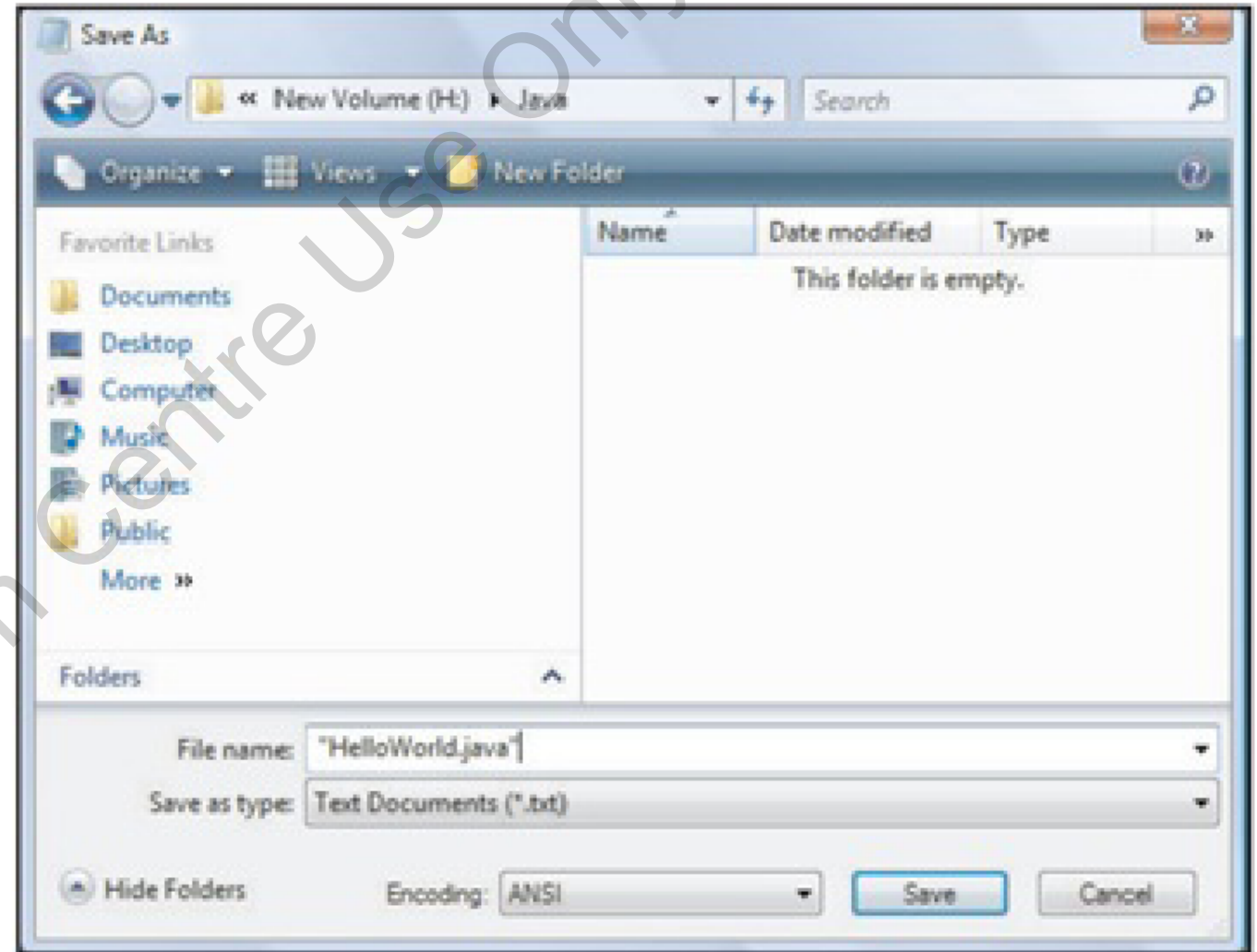
- Create a Java program
- Compile .java file
- Build and execute Java program



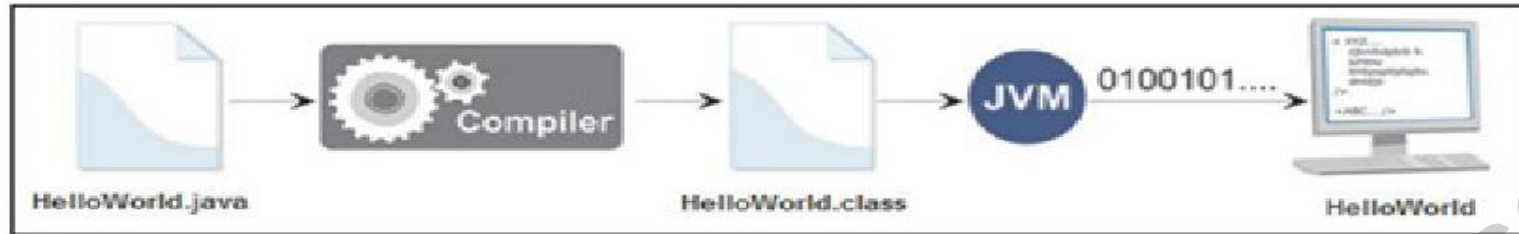
Create a Java Program



- ◆ Step 1: Class Declaration
- ◆ Step 2: Write the main method
- ◆ Step 3: Write desired functionality
- ◆ Step 4: Save the program



Compile .java File



```
Administrator: C:\Windows\system32\cmd.exe
H:\>cd Java
H:\Java>dir
Volume in drive H is New Volume
Volume Serial Number is 4266-1D52

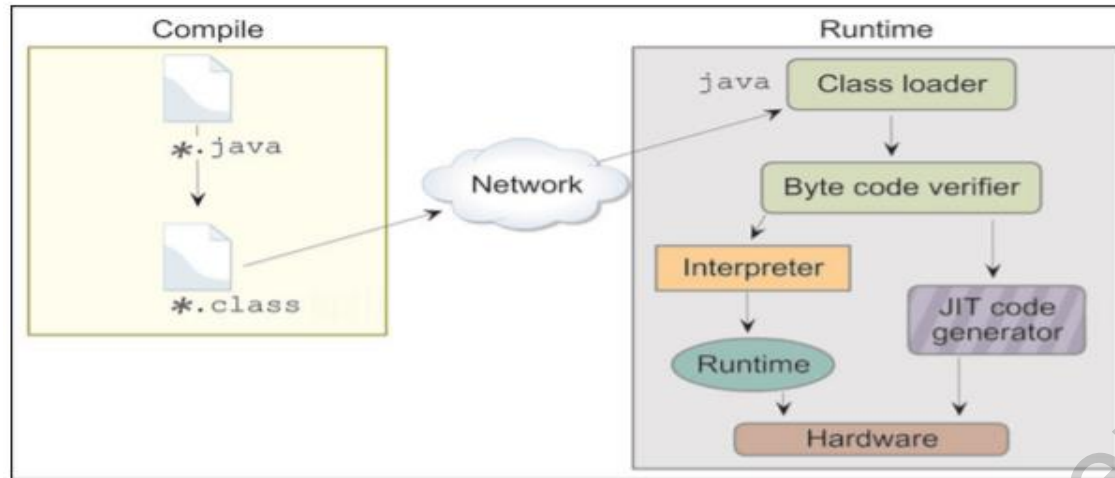
Directory of H:\Java

01/03/2020  04:42 PM    <DIR>          .
01/03/2020  04:42 PM    <DIR>          ..
01/03/2020  04:43 PM                138 HelloWorld.java
               1 File(s)                138 bytes
               2 Dir(s)  50,271,416,320 bytes free

H:\Java>
```

Option	Description
-classpath	Specifies the location for the imported classes (overrides the CLASSPATH environment variable)
-d	Specifies the destination directory for the generated class files
-g	Prints all debugging information instead of the default line number and file name
-verbose	Generates message while the class is being compiled
-version	Displays version information
sourcepath	Specifies the location of the input source file
-help	Prints a synopsis of standard options

Build and Execute Java Program



Options	Description
classpath	Specifies the location for importing classes (overrides the CLASSPATH environment variable)
-v or -verbose	Produces additional output about each class loaded and each source file compiled
-version	Displays version information and exits
-jar	Uses a JAR file name instead of a class name
-help	Displays information about help and exits
-X	Displays information about non-standard options and exits



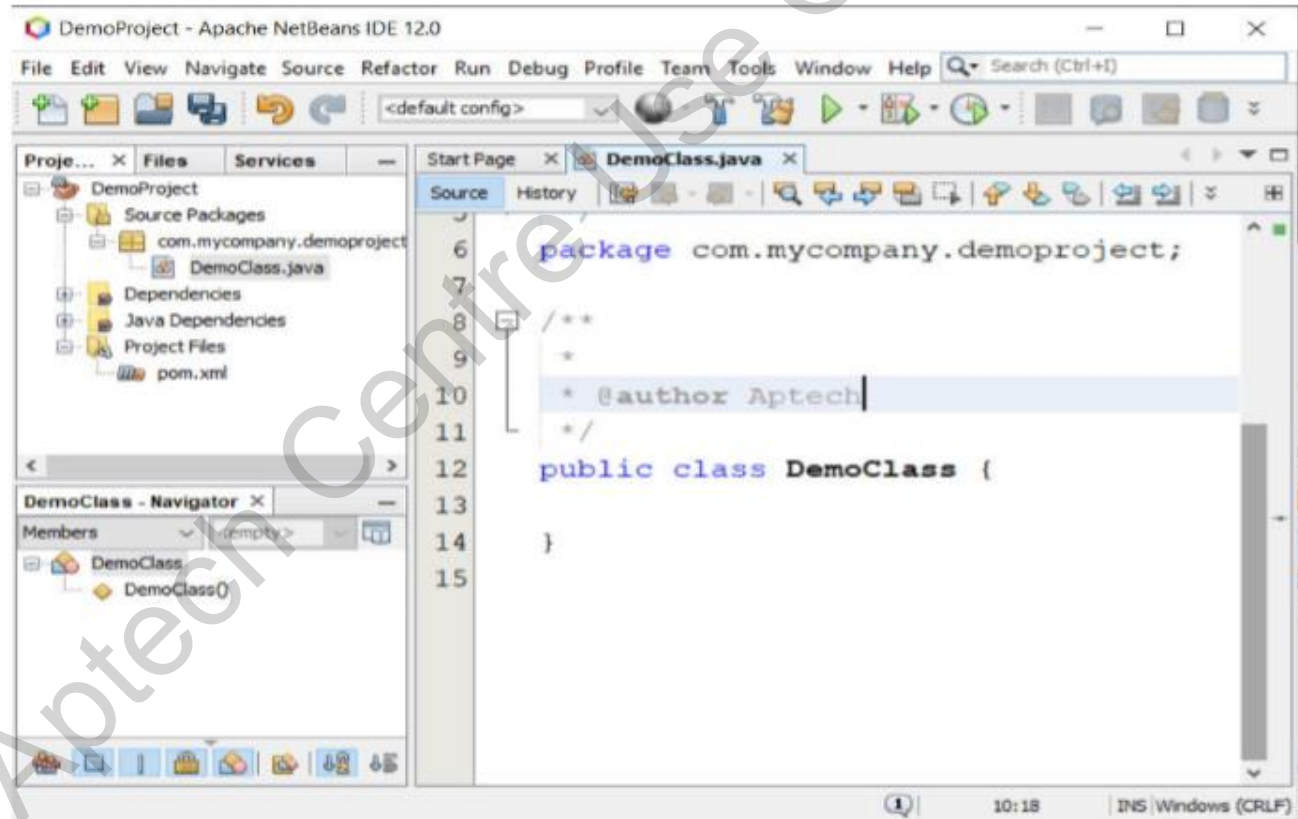
- ◆ NetBeans is an open source written purely in Java.
- ◆ It is a free and robust IDE that helps developers to create cross-platform desktop, Web, and mobile applications using Java.
- ◆ Some benefits of using NetBeans IDE for Java application development are as follows:
 - Builds IDE plug-in modules and supports rich client applications on the NetBeans platform.
 - Provides graphical user interface for building, compiling, debugging, and packaging of applications.
 - Provides simple and user-friendly IDE configuration.

For Aptech Centre Use Only

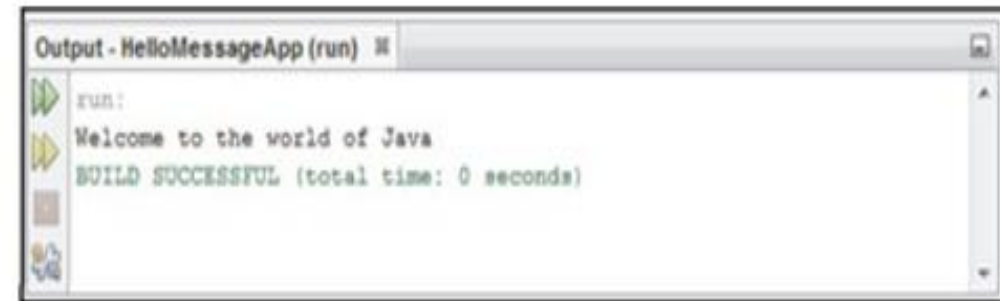
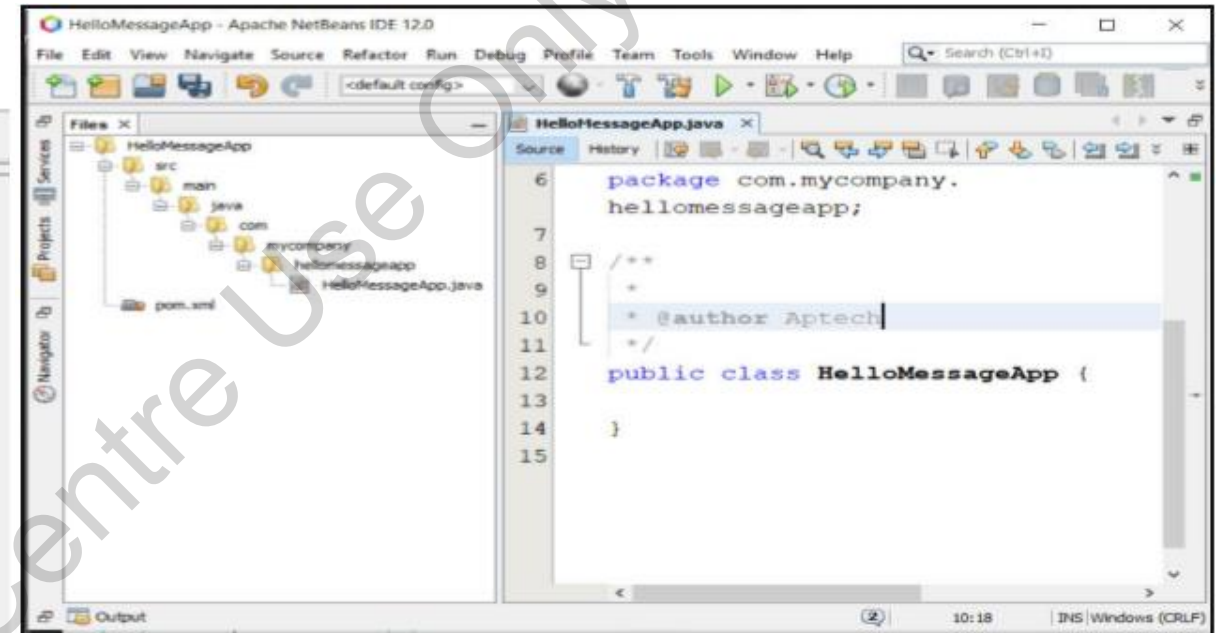
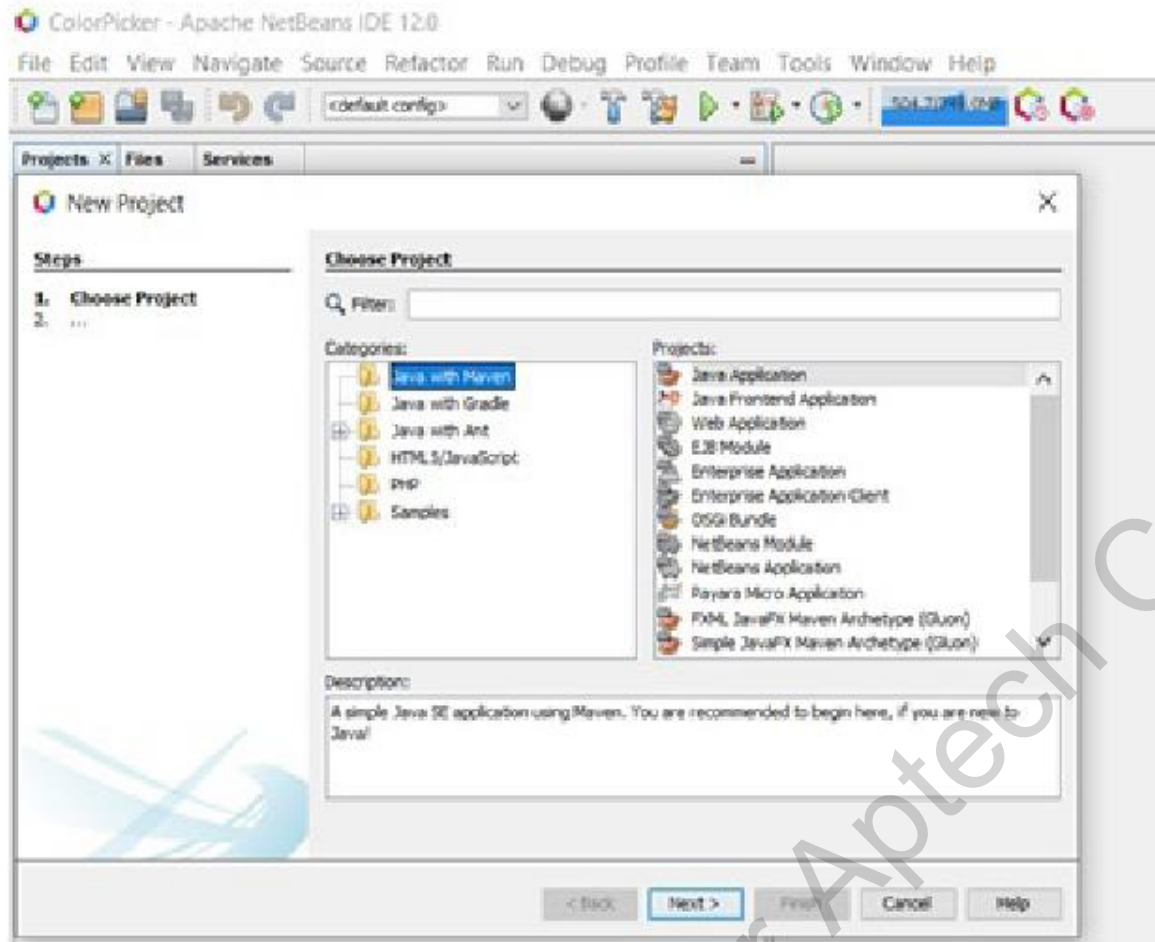


◆ The NetBeans IDE has following elements and views:

- Menu Bar
- Folders View
- Components View
- Coding and Design View
- Output View



Creating a New Project





- ◆ Comments are placed in a Java program source file.
- ◆ They are used to document the Java program and are not compiled by the compiler. They are added as remarks to make the program more readable for the user.
- ◆ A comment usually describes the operations for better understanding of the code.

There are three styles of comments supported by Java namely, single-line, multi-line, and Javadoc.

For Aptech Centre Use Only



- ◆ A single-line comment is used to document the functionality of a single line of code.

```
class HelloWorld {  
    public static void main(String[] args) {  
        //The println() method is used to display a message on the screen  
        System.out.println("Welcome to the world of Java");  
    }  
}
```

- ◆ There are two ways of using single-line comments that are as follows:

Beginning-of-line comment - This type of comment can be placed before the code (on a different line).

End-of-line comment - This type of comment is placed at the end of the code (on the same line). Conventions for using single-line comments are as follows:

- Insert a space after the forward slashes.
- Capitalize the first letter of the first word.



- ◆ A multi-line comment is a comment that spans multiple lines.
- ◆ A multi-line comment starts with a forward slash and an asterisk (/*).
- ◆ It ends with an asterisk and a forward slash (*).
- ◆ Anything that appears between these delimiters is considered to be a comment.

For Apteck Centre Use Only



- ◆ A Javadoc comment is used to document public or protected classes, attributes, and methods.
- ◆ It starts with `/**` and ends with `*/`.
- ◆ Everything between the delimiters is a comment, even if it spans multiple lines.
- ◆ The `javadoc` command can be used for generating Javadoc comments.

For Aptech Centre Use Only



- ◆ The development of application software is performed using a programming language that enforces a particular style of programming, also referred to as programming paradigm.
- ◆ In structured programming paradigm, the application development is decomposed into a hierarchy of subprograms.
- ◆ In object-oriented programming paradigm, applications are designed around data, rather than focusing only on the functionalities.
- ◆ The main building blocks of an OOP language are classes and objects. An object represents a real-world entity and a class is a conceptual model.
- ◆ Java is an OOP language as well a platform used for developing applications that can be executed on different platforms.
- ◆ Hidden classes, Z Garbage Collector, Sealed classes, Records, and improved security with Edwards-Curve Digital Signature algorithm are some of the new features in Java 15.
- ◆ Apache NetBeans IDE version 12 and higher provide an integrated development environment to create, compile, and execute Java programs.
- ◆ The components of Java SE platform are JDK and JRE. JRE provides JVM and Java libraries that are used to run a Java program. JDK includes the necessary development tools, runtime environment, and APIs for creating Java programs.