UNIT – I

INTRODUCTION TO JAVA

1.1 Basics of Java

- 1.1.1 History of Java
- Java was developed by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991. It took 18 months to develop the first working version.

- This language was initially called "**Oak**" but was renamed "**Java**" in 1995.

- The primary motivation was the need for a **platform-independent** language. The trouble with C and C++ (and most other languages) is that they are designed to be compiled for a specific target.

- Java inherits the **syntax of C** and many of the **object-oriented features of C++.**

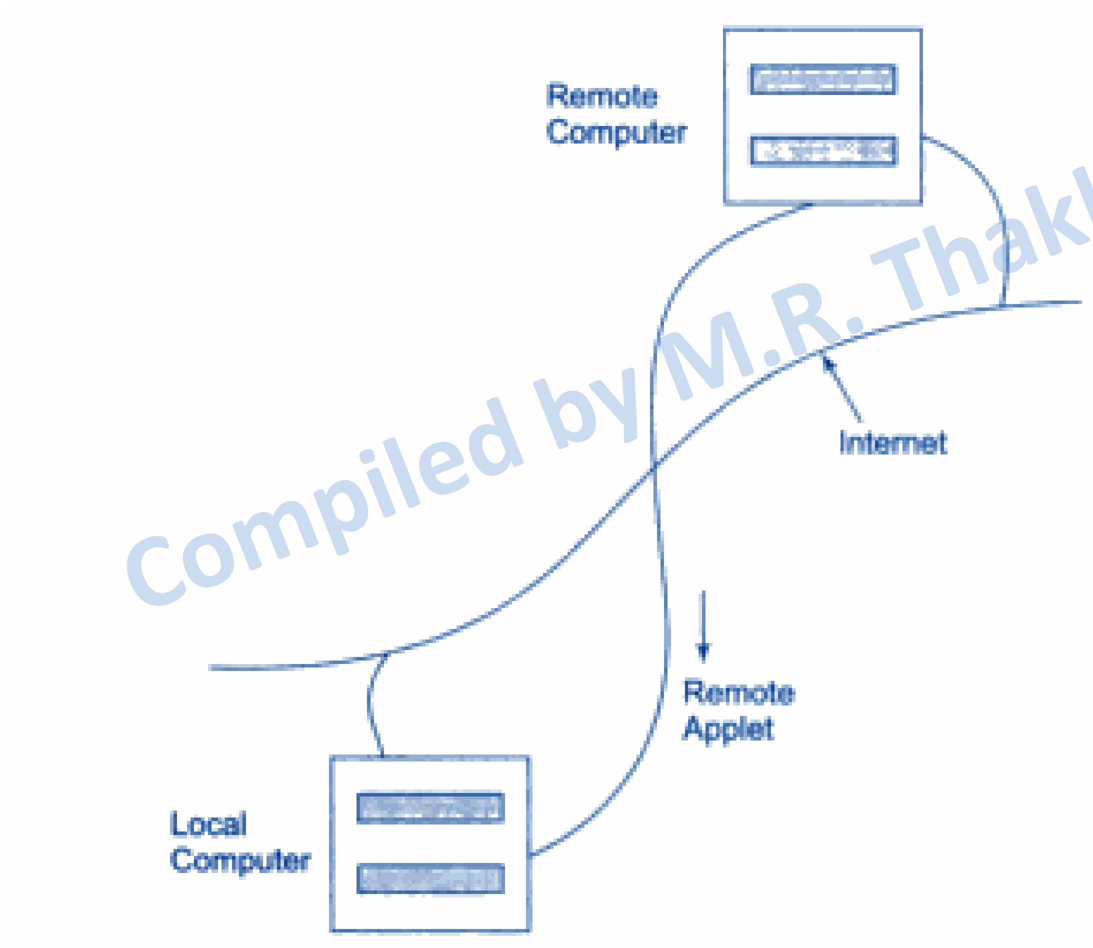
1.1 Basics of Java

- 1.1.2 Java and Internet

- Java can be used to create two types of programs: **applications** and **applets**.
- An application is a program that **runs on your computer**, under the operating system of that computer.
- An applet is an application designed **to be transmitted over the Internet** and executed by a Java-compatible **Web browser**.
- An applet is actually a small Java program, dynamically downloaded across the network, just like an image, sound file, or video clip.
- The important difference is that an applet is **an intelligent program**, not just an animation or media file. In other words, an applet is a program that can **react to user input and dynamically change**—not just run the same animation or sound over and over.

1.1 Basics of Java

- 1.1.2 Java and Internet



1.1 Basics of Java

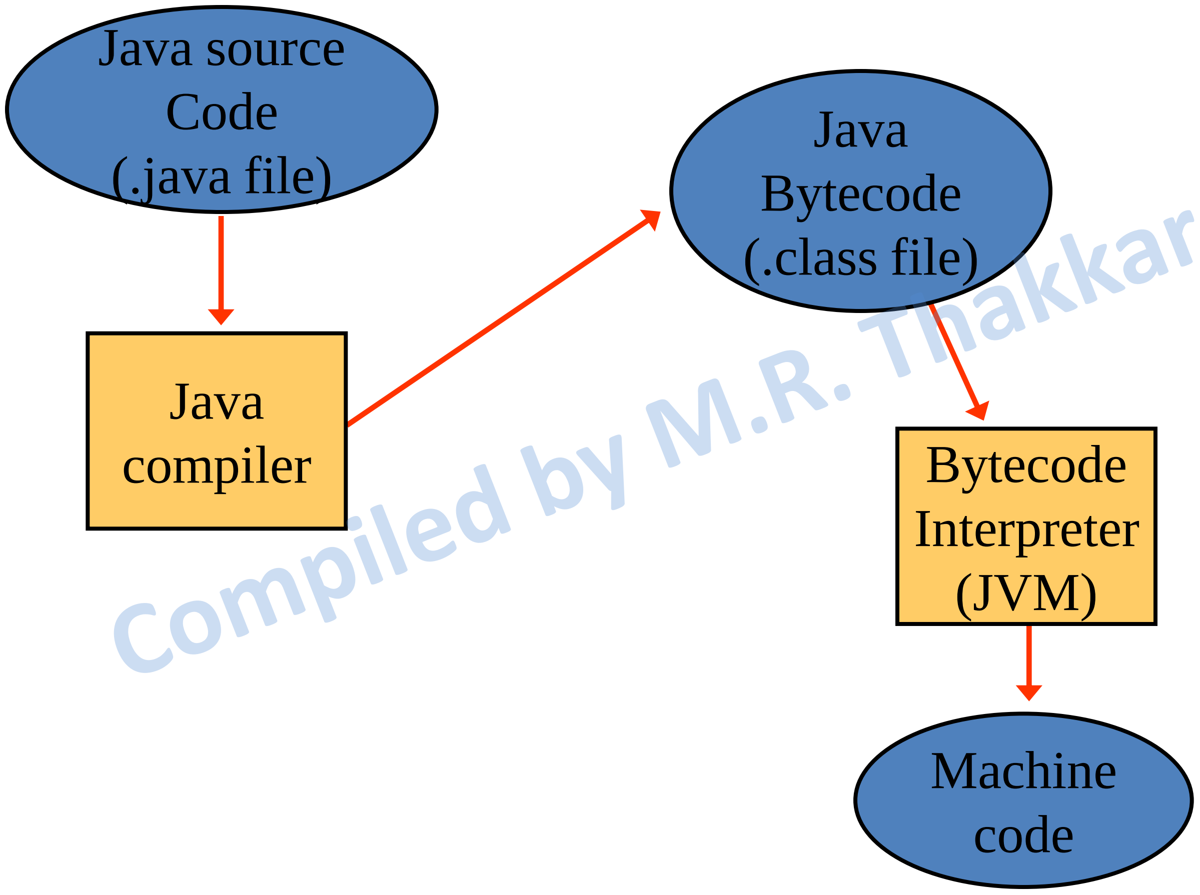
- 1.1.3 Advantages of Java

- Advantages of Java are as below:
  - Simple
  - Secure
  - Portable
  - Object-oriented
  - Robust
  - Multithreaded
  - Architecture-neutral
  - Interpreted
  - Distributed
  - Dynamic

1.2 Java Virtual Machine & Bytecode

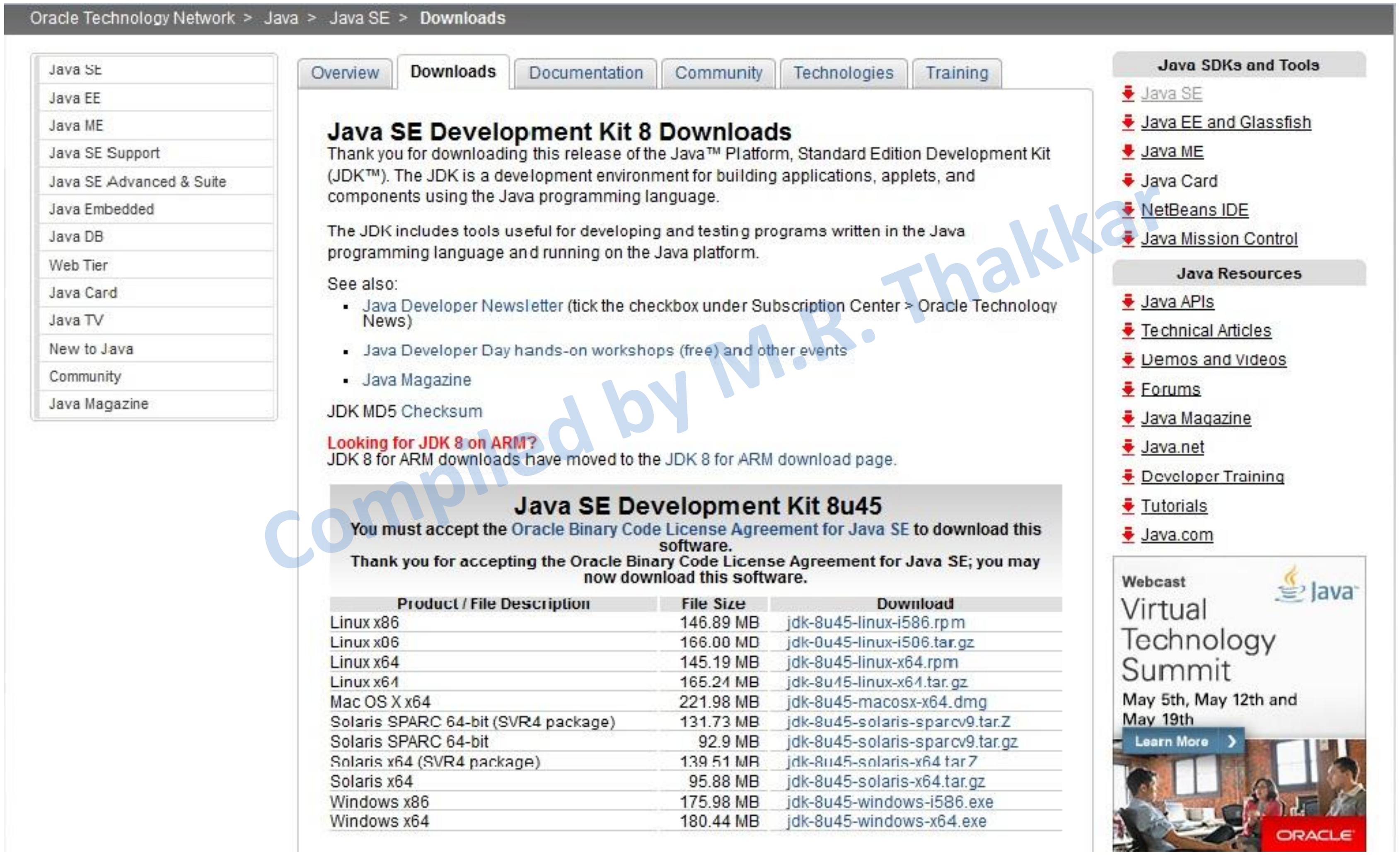
- The Java compiler translates Java source code into a special representation called **bytecode**.
- Java bytecode is **not the machine language** for any traditional CPU.
- An interpreter (JVM), translates bytecode into machine language and executes it.

1.2 Java Virtual Machine & Bytecode


Java source Code (.java file)
Java compiler
Java Bytecode (.class file)
Bytecode Interpreter (JVM)
Machine code

1.3 Java Environment Setup

- http://www.oracle.com/technetwork/java/javase/downloads/index.html



Oracle Technology Network > Java > Java SE > Downloads

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter (tick the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK MD5 Checksum

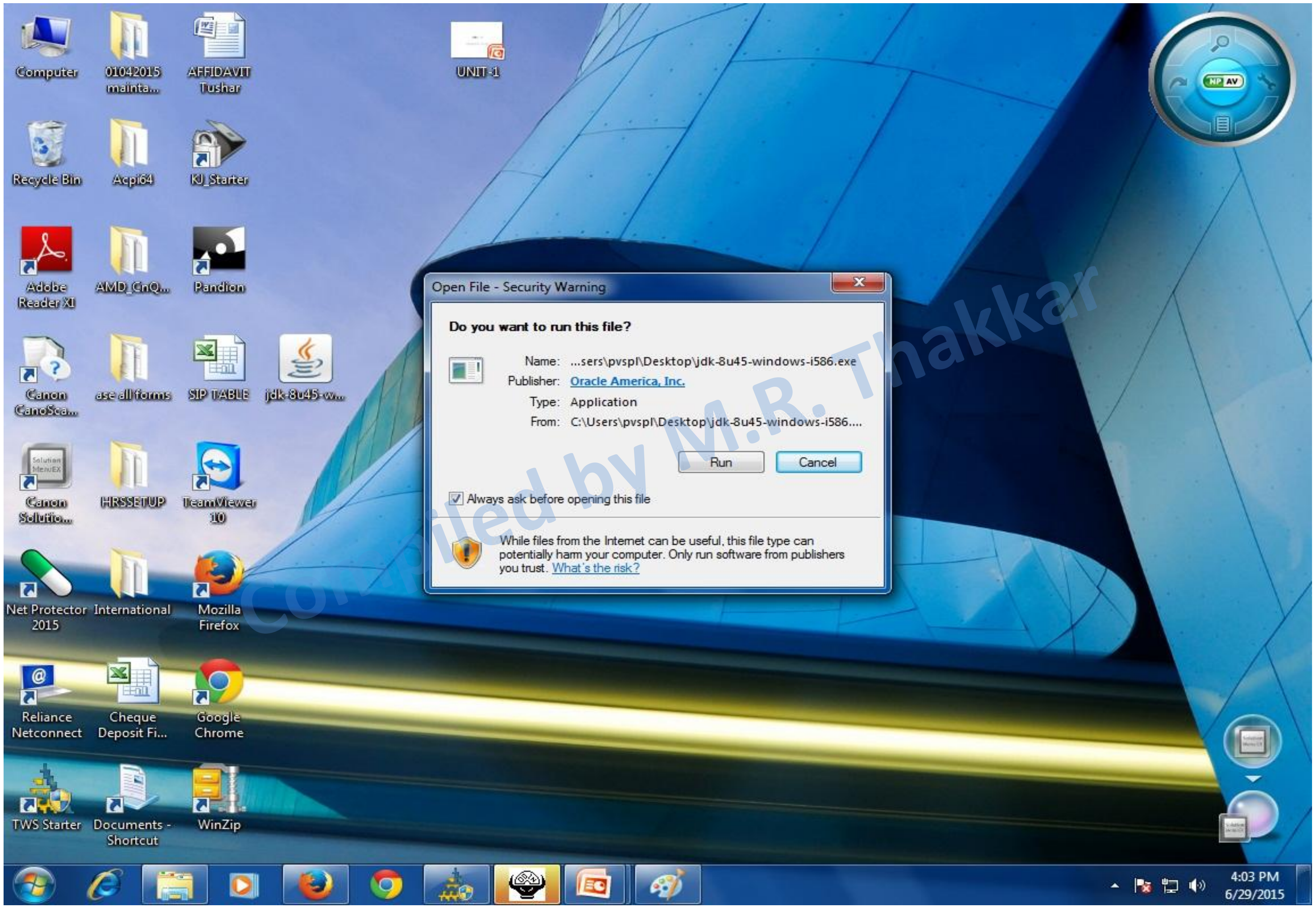
Looking for JDK 8 on ARM?
JDK 8 for ARM downloads have moved to the JDK 8 for ARM download page.

Java SE Development Kit 8u45

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	146.89 MB	jdk-8u45-linux-i586.rpm
Linux x86	166.00 MB	jdk-8u45-linux-i586.tar.gz
Linux x64	145.19 MB	jdk-8u45-linux-x64.rpm
Linux x64	165.24 MB	jdk-8u45-linux-x64.tar.gz
Mac OS X x64	221.98 MB	jdk-8u45-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	131.73 MB	jdk-8u45-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	92.9 MB	jdk-8u45-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	139.51 MB	jdk-8u45-solaris-x64.tar.Z
Solaris x64	95.88 MB	jdk-8u45-solaris-x64.tar.gz
Windows x86	175.98 MB	jdk-8u45-windows-i586.exe
Windows x64	180.44 MB	jdk-8u45-windows-x64.exe

1.3 Java Environment Setup



Open File - Security Warning

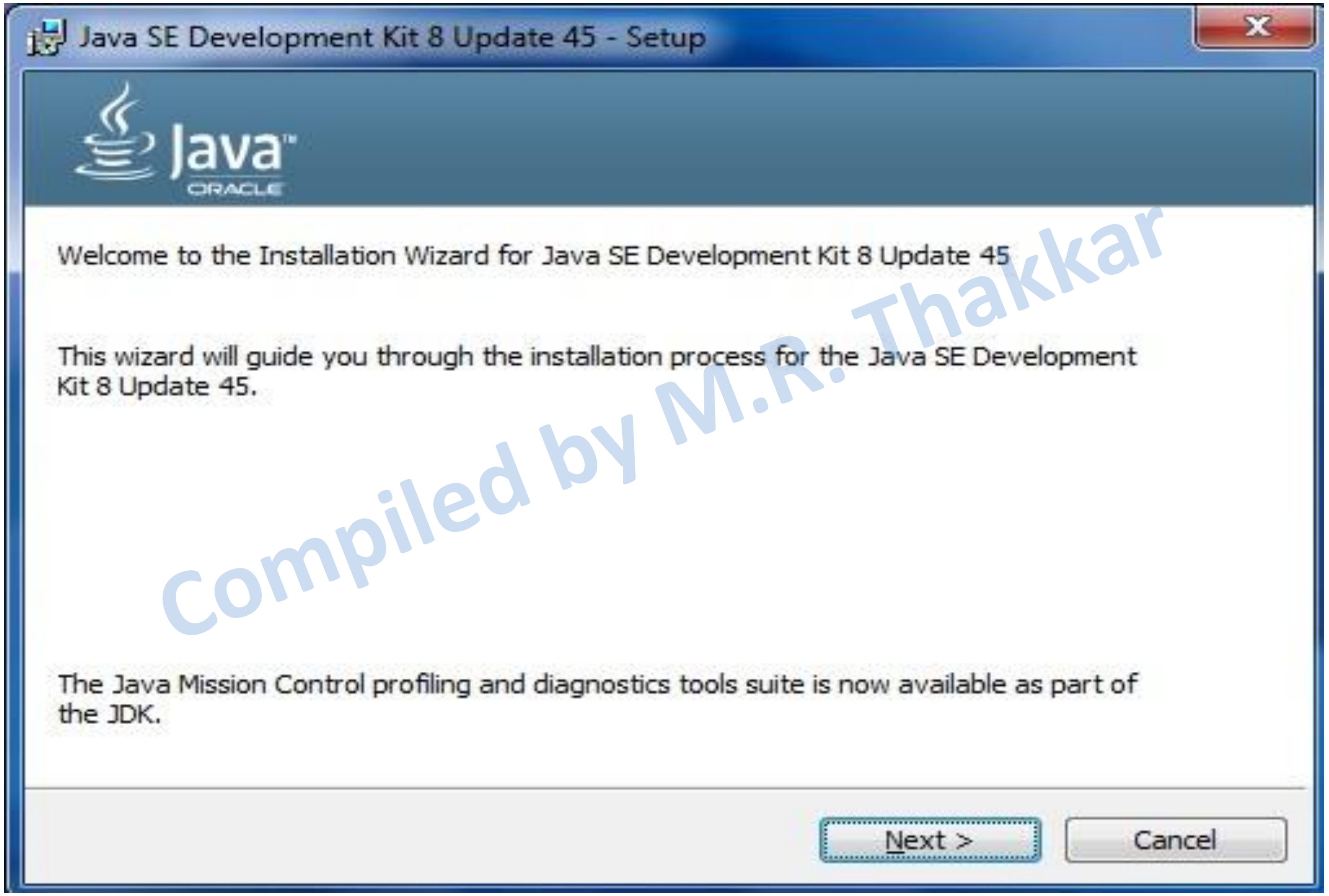
Do you want to run this file?

Name: ...sers\pvspl\Desktop\jdk-8u45-windows-i586.exe
Publisher: Oracle America, Inc.
Type: Application
From: C:\Users\pvspl\Desktop\jdk-8u45-windows-i586....

[✓] Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

1.3 Java Environment Setup



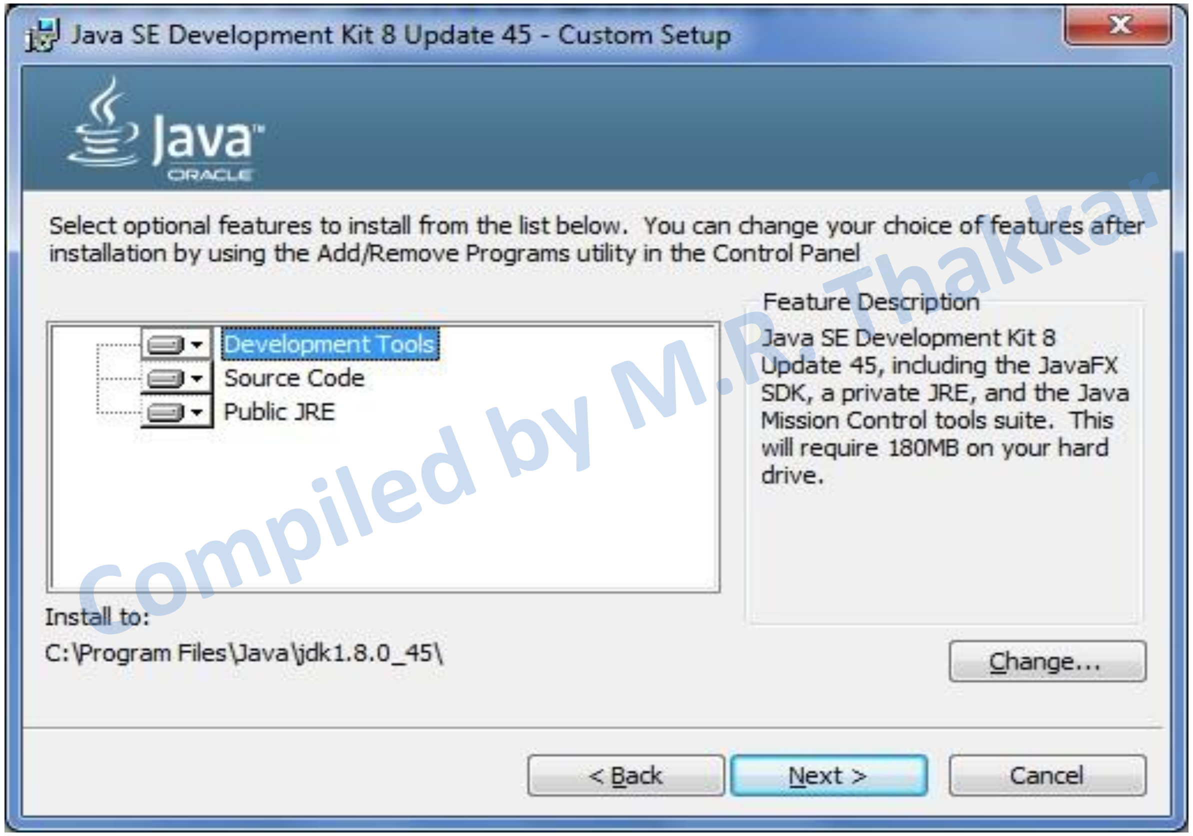
Java SE Development Kit 8 Update 45 - Setup

Welcome to the Installation Wizard for Java SE Development Kit 8 Update 45

This wizard will guide you through the installation process for the Java SE Development Kit 8 Update 45.

The Java Mission Control profiling and diagnostics tools suite is now available as part of the JDK.

1.3 Java Environment Setup



Java SE Development Kit 8 Update 45 - Custom Setup

Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

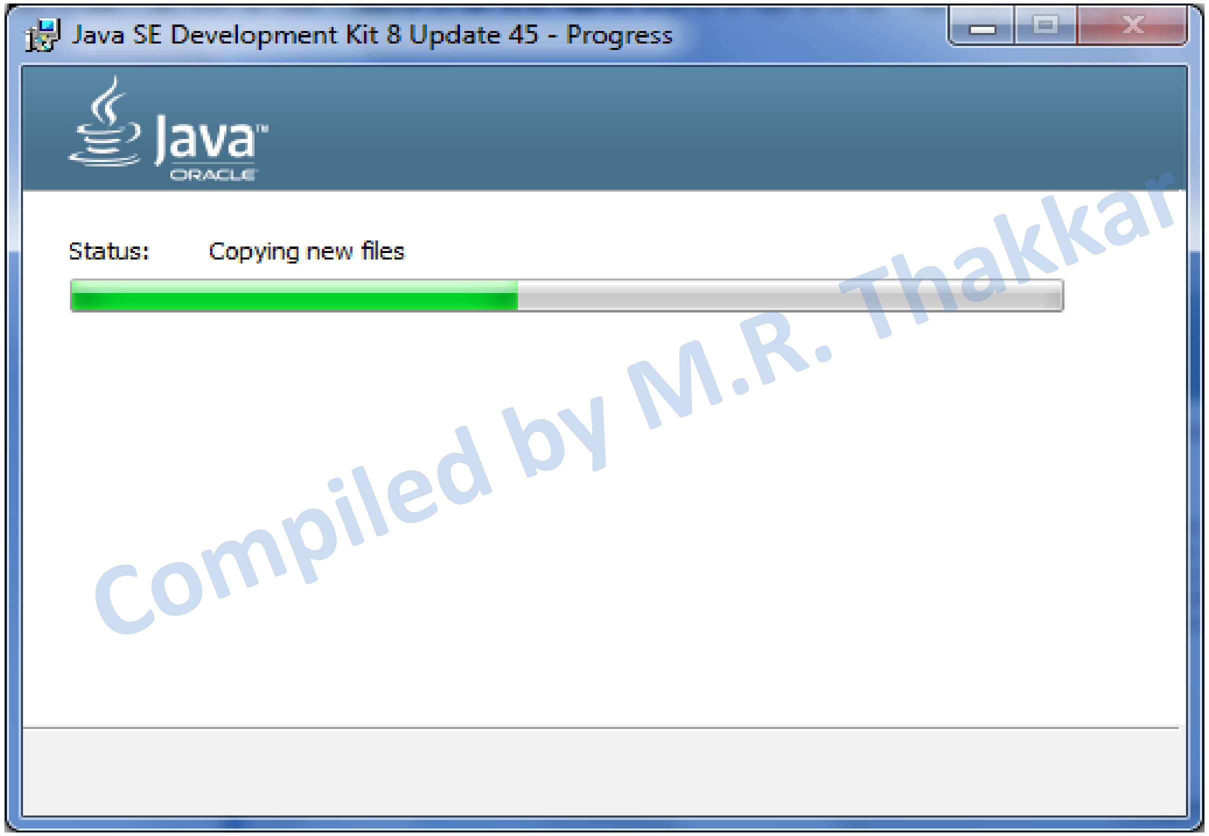
Feature Description

Java SE Development Kit 8 Update 45, including the JavaFX SDK, a private JRE, and the Java Mission Control tools suite. This will require 180MB on your hard drive.

Install to:

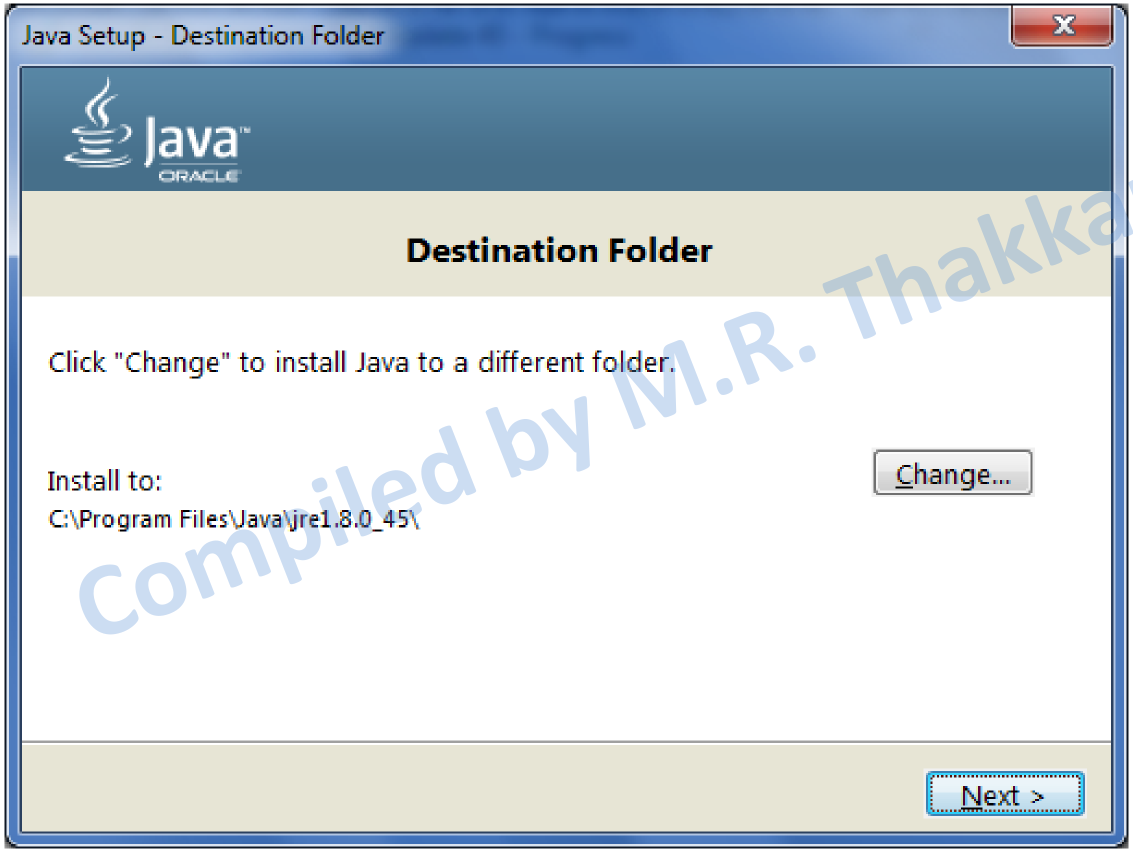
C:\Program Files\Java\jdk1.8.0_45\

1.3 Java Environment Setup


Java SE Development Kit 8 Update 45 - Progress

Status: Copying new files

1.3 Java Environment Setup

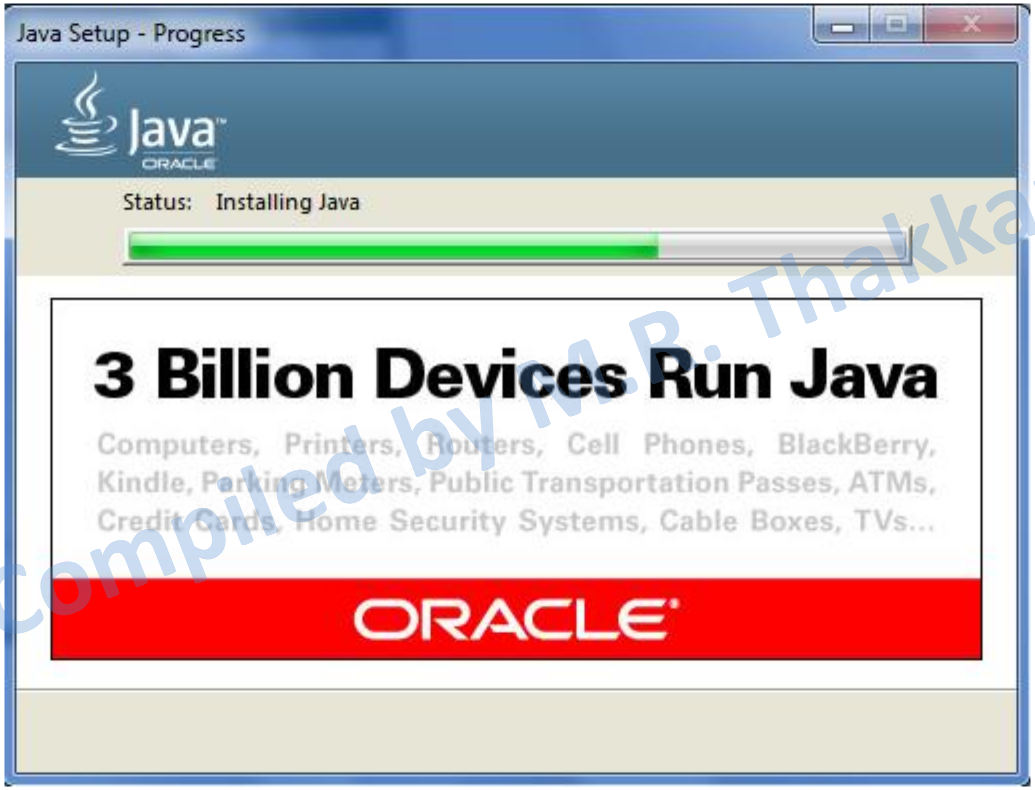


Destination Folder

Click "Change" to install Java to a different folder.

Install to:
C:\Program Files\Java\jre1.8.0_45\

1.3 Java Environment Setup

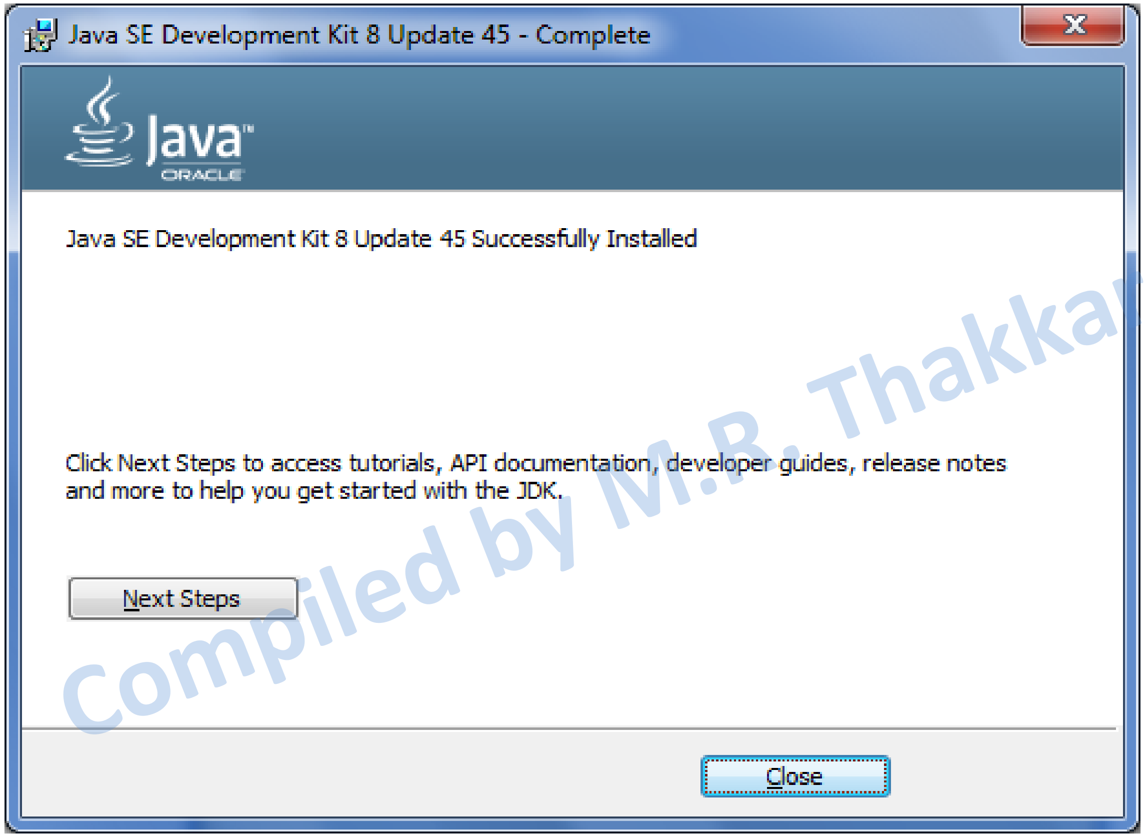


Status: Installing Java

3 Billion Devices Run Java

Computers, Printers, Routers, Cell Phones, BlackBerry, Kindle, Parking Meters, Public Transportation Passes, ATMs, Credit Cards, Home Security Systems, Cable Boxes, TVs...

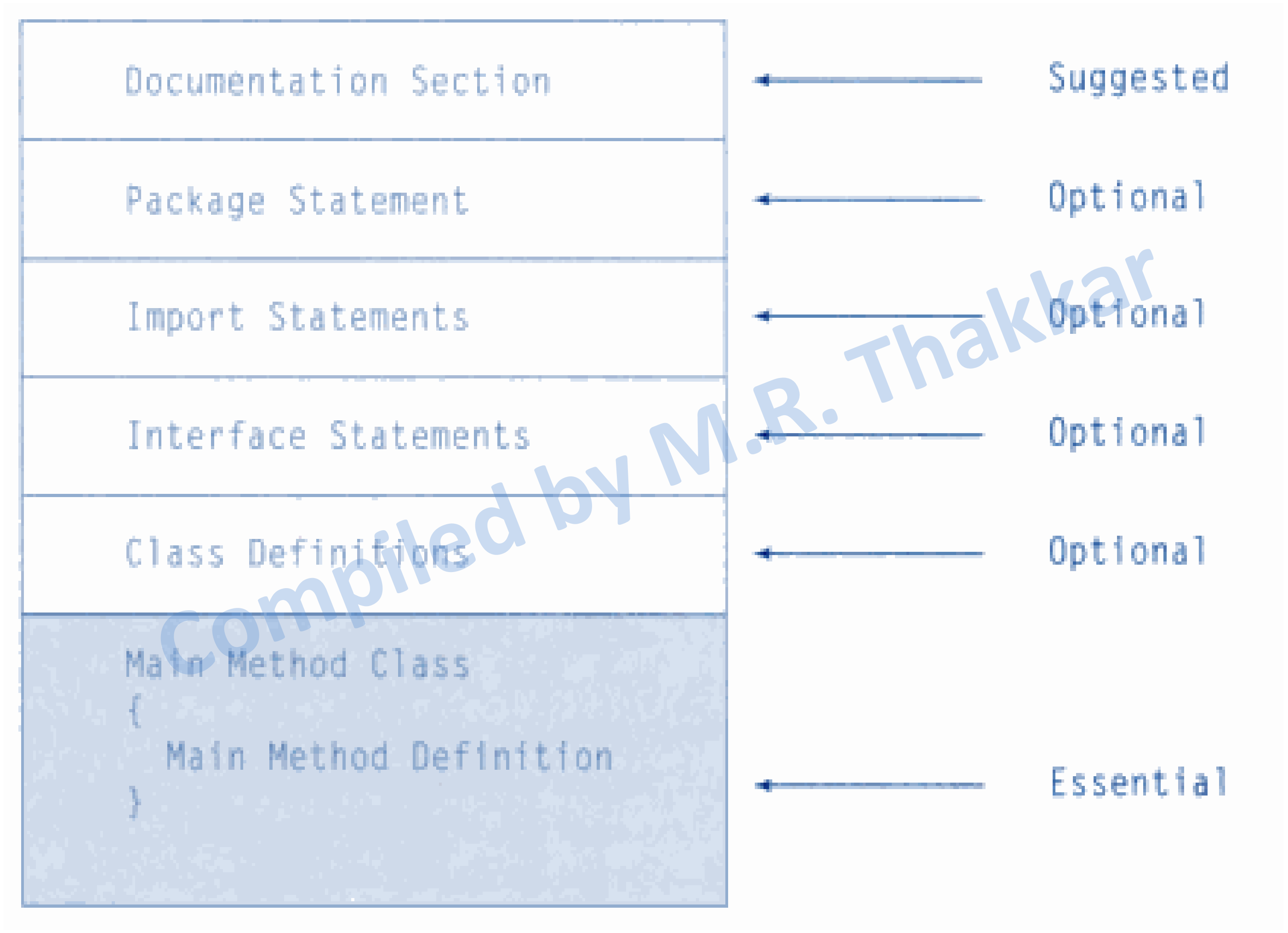
1.3 Java Environment Setup


Java SE Development Kit 8 Update 45 - Complete

Java SE Development Kit 8 Update 45 Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

1.4 Java Program Structure



1.4 Java Program Structure

- **DOCUMENTATION SECTION**: It is a set of comment lines giving the details about the program, details about the author and the other details,which the programmer would like to include.

- **PACKAGE STATEMENT**: This statement declares a package and informs the compiler that the classes defined here belong to this package.

  Example : package student;

- **IMPORT STATEMENT**: Using import statement ,we can access to classes that part of other named packages.

  Example : import student.test;

  – This statement instructs the interpreter to load the test class contained in package student.

1.4 Java Program Structure

- **INTERFACE STATEMENT**: An interface is like a class but includes group of methods declaration.
- **CLASS DEFINITION**: Java program may contain multiple class definition.
- **MAIN METHOD CLASS**: Java application program requires main method as starting point. There must be one class with main method in java program.

1.5 POP Vs OOP

- 1.5.1 POP

- POP, the problem is viewed as a set of tasks. A set of functions are written to accomplish this tasks.
- In POP, primary focus is on fuctions.
- A typical program structure of the POP is as shown in below fig:

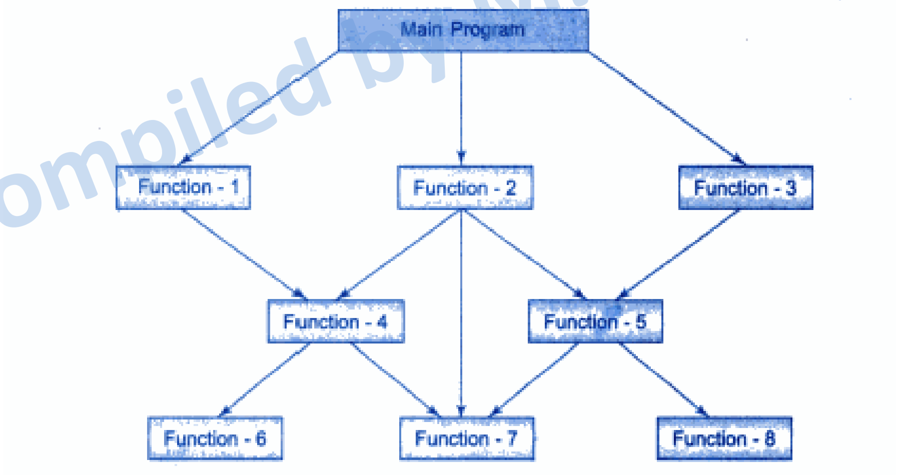


Fig. 1.4 ⇔ Typical structure of procedure-oriented programs

1.5 POP Vs OOP

- 1.5.2 OOP

- OOP allows decomposition of the problems into set of entities called objects and then builds the data and functions around these objects.
- OOP treats data as a critical element and does not allow it to flow freely around the system. It ties data more closely to the functions, that operate on it.

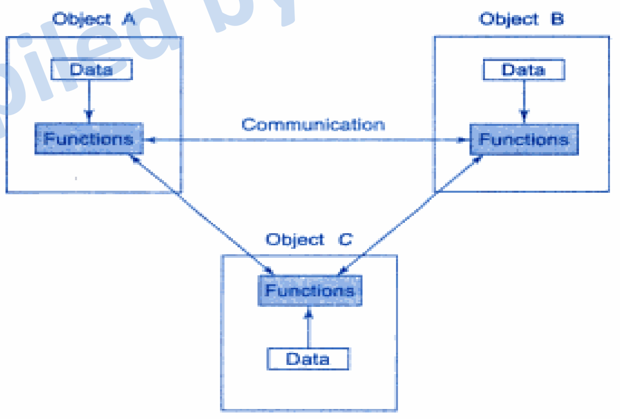
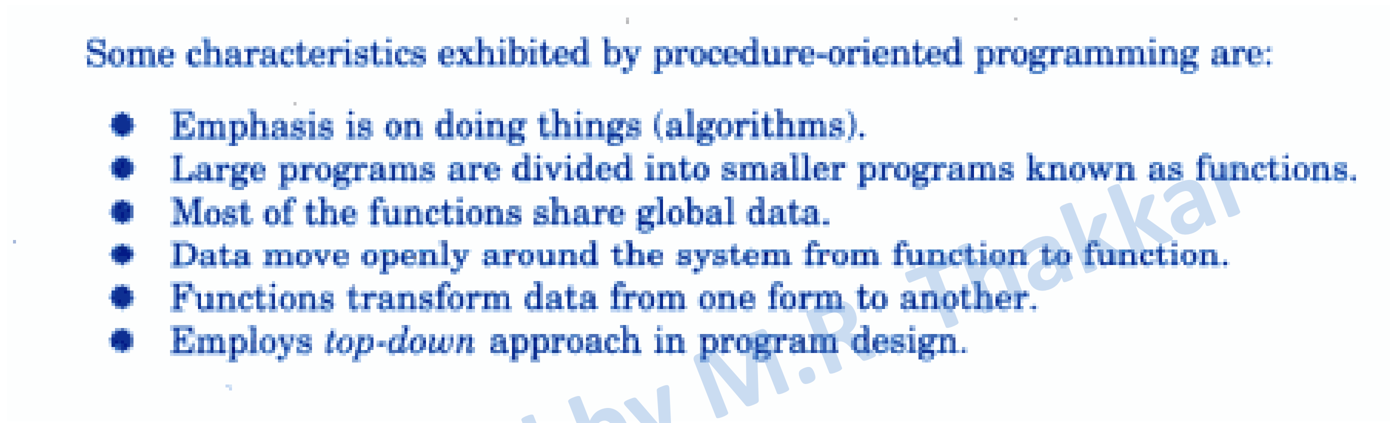


Fig. 1.6 ⇔ Organization of data and functions in OOP

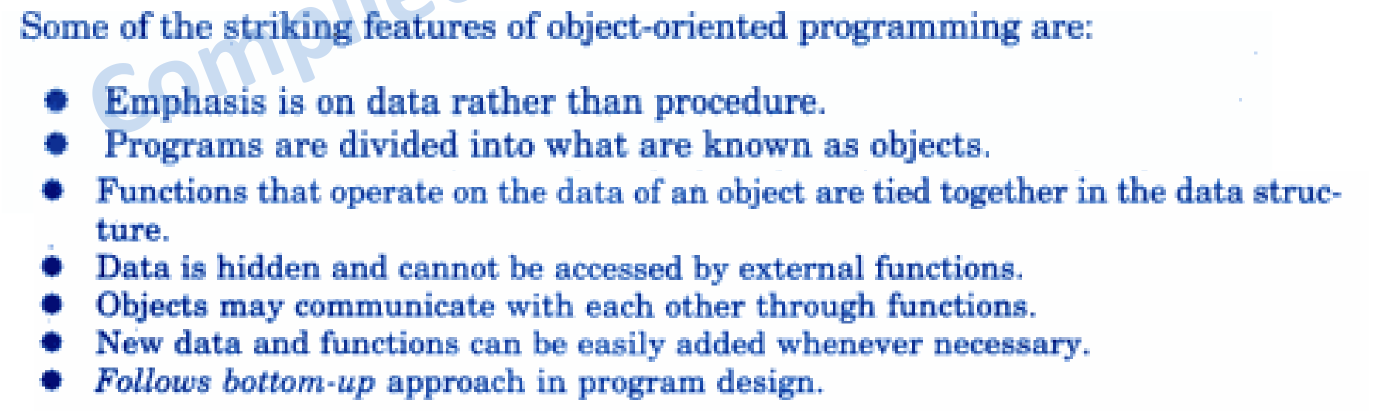
1.5 POP Vs OOP

- 1.5.3 Difference



Some characteristics exhibited by procedure-oriented programming are:

- Emphasis is on doing things (algorithms).
- Large programs are divided into smaller programs known as functions.
- Most of the functions share global data.
- Data move openly around the system from function to function.
- Functions transform data from one form to another.
- Employs *top-down* approach in program design.



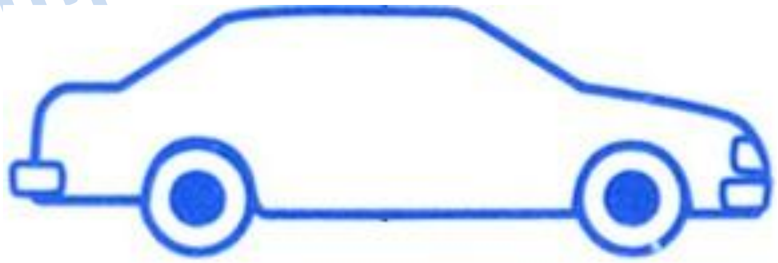
Some of the striking features of object-oriented programming are:

- Emphasis is on data rather than procedure.
- Programs are divided into what are known as objects.
- Functions that operate on the data of an object are tied together in the data structure.
- Data is hidden and cannot be accessed by external functions.
- Objects may communicate with each other through functions.
- New data and functions can be easily added whenever necessary.
- *Follows bottom-up* approach in program design.

1.6 Basics of OOP

- 1.6.1 Abstraction

- Abstraction refers to act of **representing essential features without including the background details**.

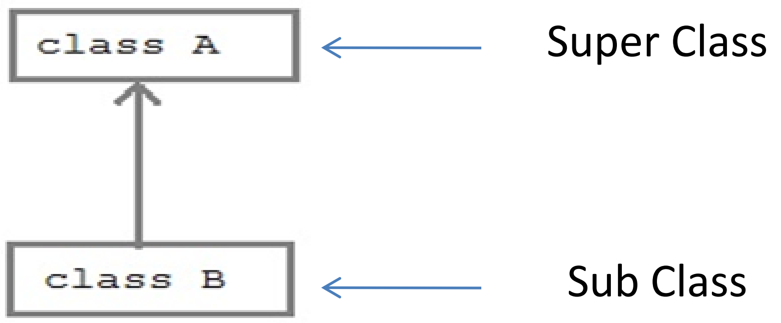


1.6 Basics of OOP

- 1.6.2 Inheritance

- Inheritance is the process by which one class, **acquire the properties** of another class. The old class is known as the **super class** and new class is known as the **sub class**.

- In OOP, the concept of inheritance provides the idea of **reusability**. This means that we can add additional information to an existing class without modifying it.

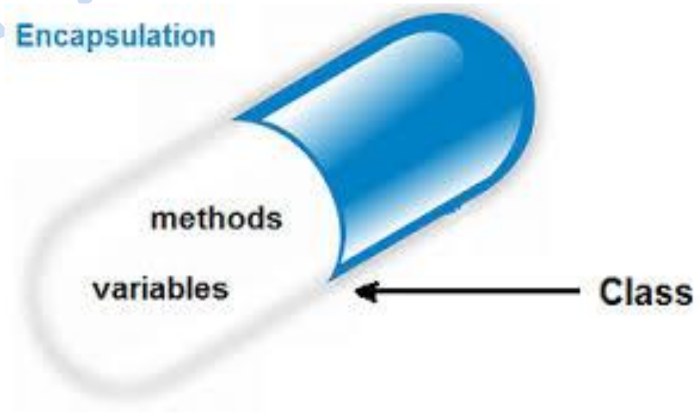


1.6 Basics of OOP

- 1.6.3 Encapsulation

- The **wrapping of the data and function together** into a single unit class, is known as encapsulation.

```
class  Test
{
        int  i;

        void  display()
        {
                System.out.print(" i = " + i);
        }
}
```



1.6 Basics of OOP

- 1.6.4 Classes

- Class is **user defined data type** and behaves like the built-in types of a programming language. **Objects are variable of type class**.
- Once a class is defined, we can create number of objects belonging to that class.
- For example, if Fruit has been defined as a class, then statement :

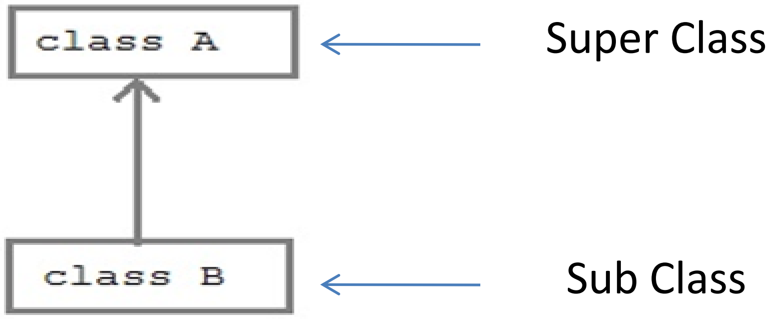
  Fruit mango = new Fruit();
- Will create an object mango, belonging to the class Fruit.

1.6 Basics of OOP

- 1.6.5 Subclass & Superclass

- Inheritance is the process by which one class, **acquire the properties** of another class.

- The old class is known as the **super class** and new class is known as the **sub class**.



1.6 Basics of OOP

- 1.6.6 Polymorphism
- Polymorphism means **the ability to take more than one form.**

- **For example**:
- Using same function name to perform different types of tasks is known as function overloading which is example of the polymorphism.

```
class  A
{
          void display ()
          {
                    System.out.println("No parameters");
          }

          void display (int a)
          {
                    System.out.println("a: " + a);
          }
}
```

1.6 Basics of OOP

- 1.6.7 Overloading

- Using single method name **to perform different types of tasks** is known as method overloading which is example of the polymorphism.

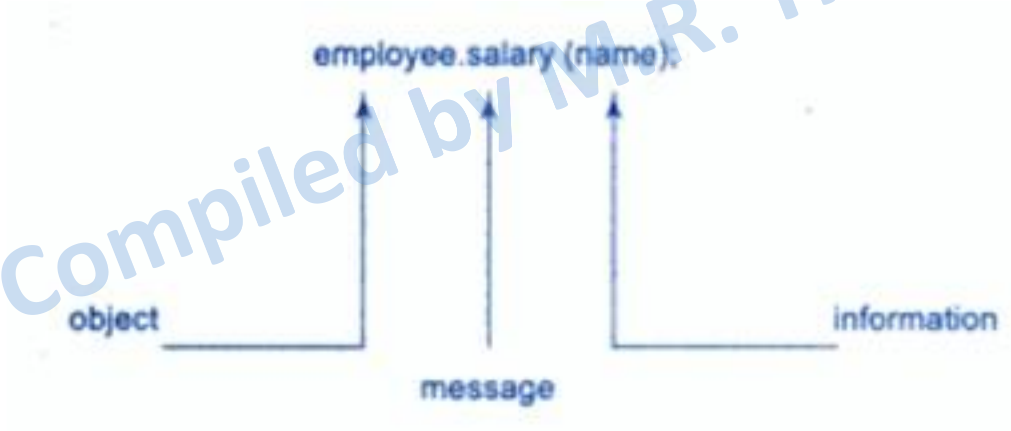
```
void add(int a, int b);
void add(float a, float b);
```

- Here first add function is used to perform addition of two integer numbers, while second add function is used to perform addition of two float numbers.

1.6 Basics of OOP

- 1.6.8 Message Communication

- Objects communicate with each other by sending and receiving the information, in the same way as people pass message to one another.

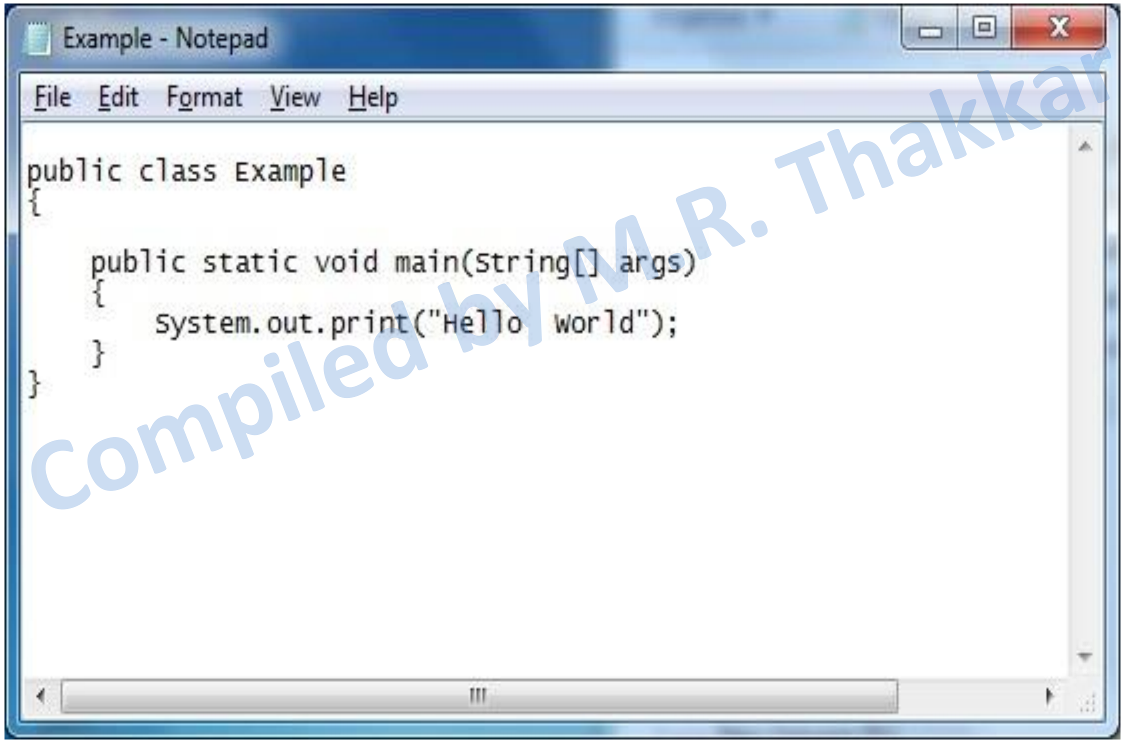


1.7 Hello World Program

```
public class Example
{
     Public static void main(String args[])
    {
         System.out.print(“Hello World”);
    }


}
```

1.7 Hello World Program

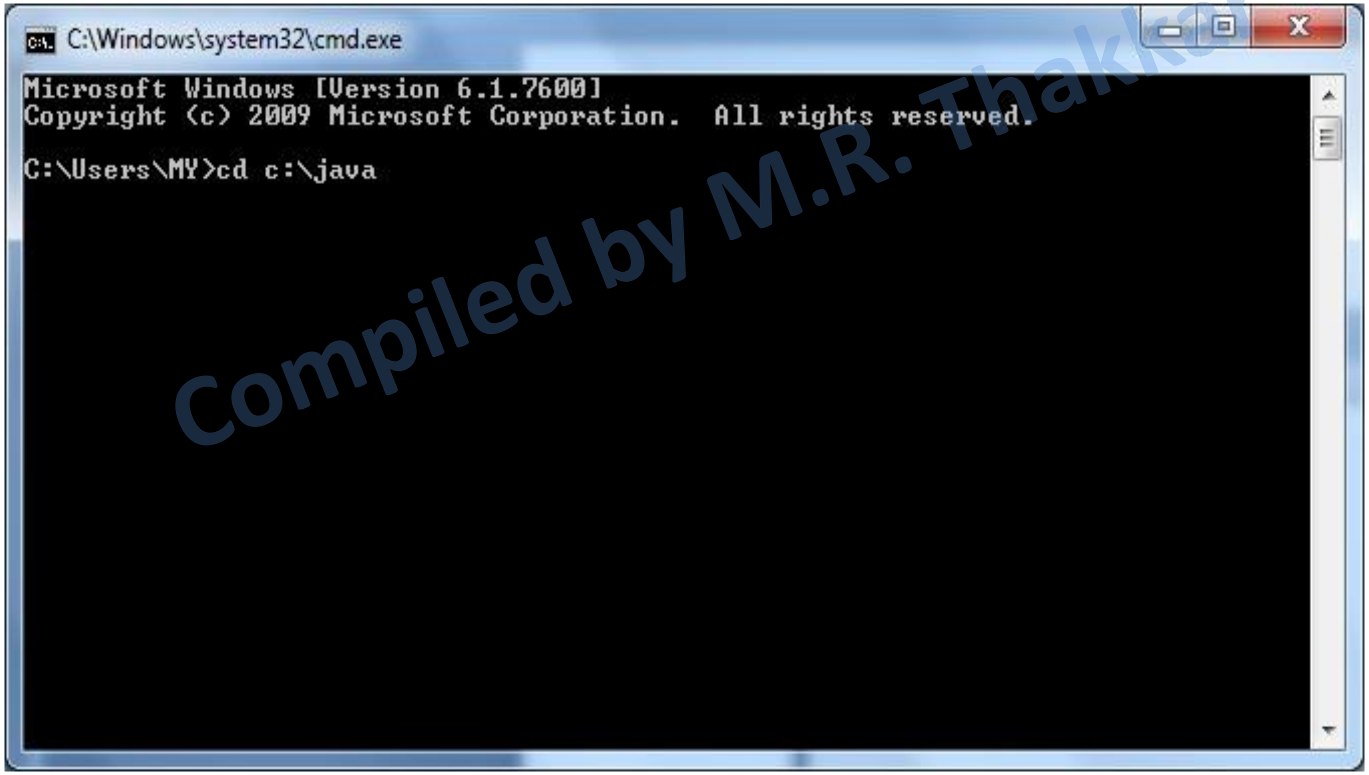
➢ Write the Program code in notepad and save the file as **Example.java**



```
public class Example
{
    public static void main(String[] args)
    {
        System.out.print("Hello  World");
    }
}
```

1.7 Hello World Program

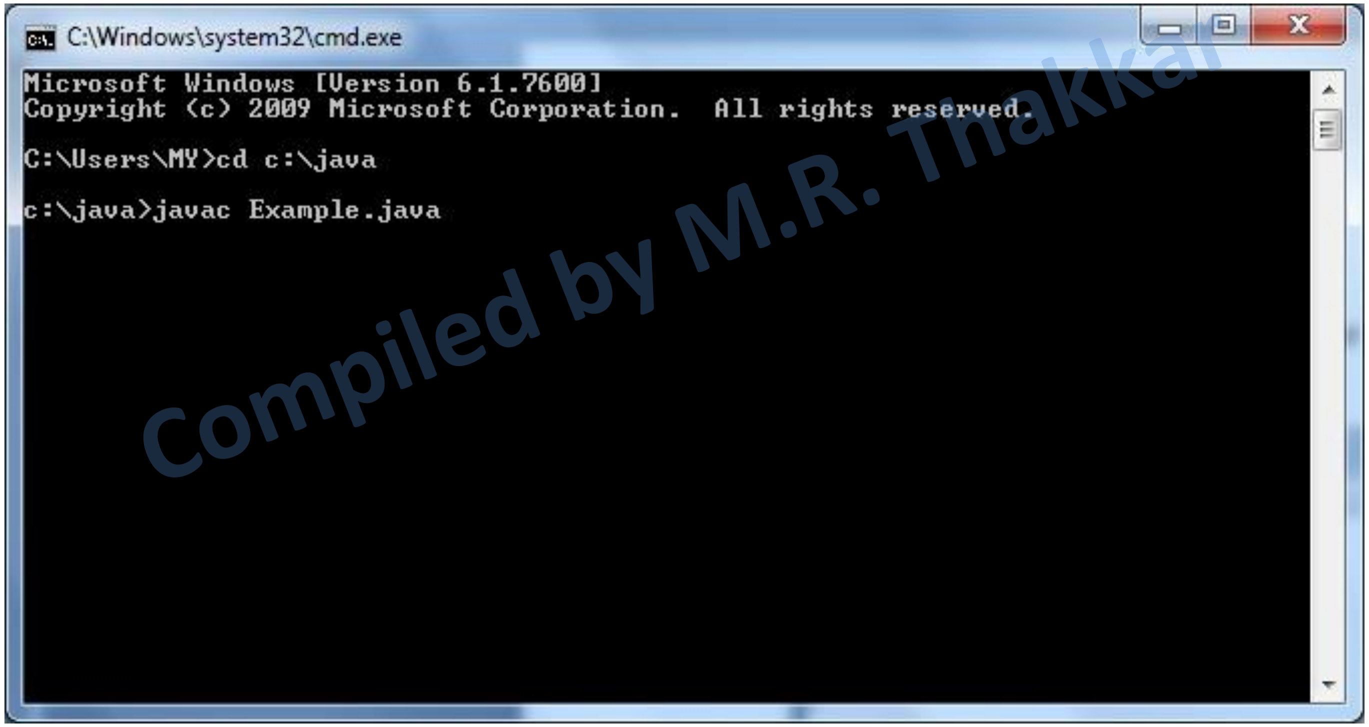
- Open Command Prompt.
- **Change the directory, where Example.java file is stored.**



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\MY>cd c:\java
```

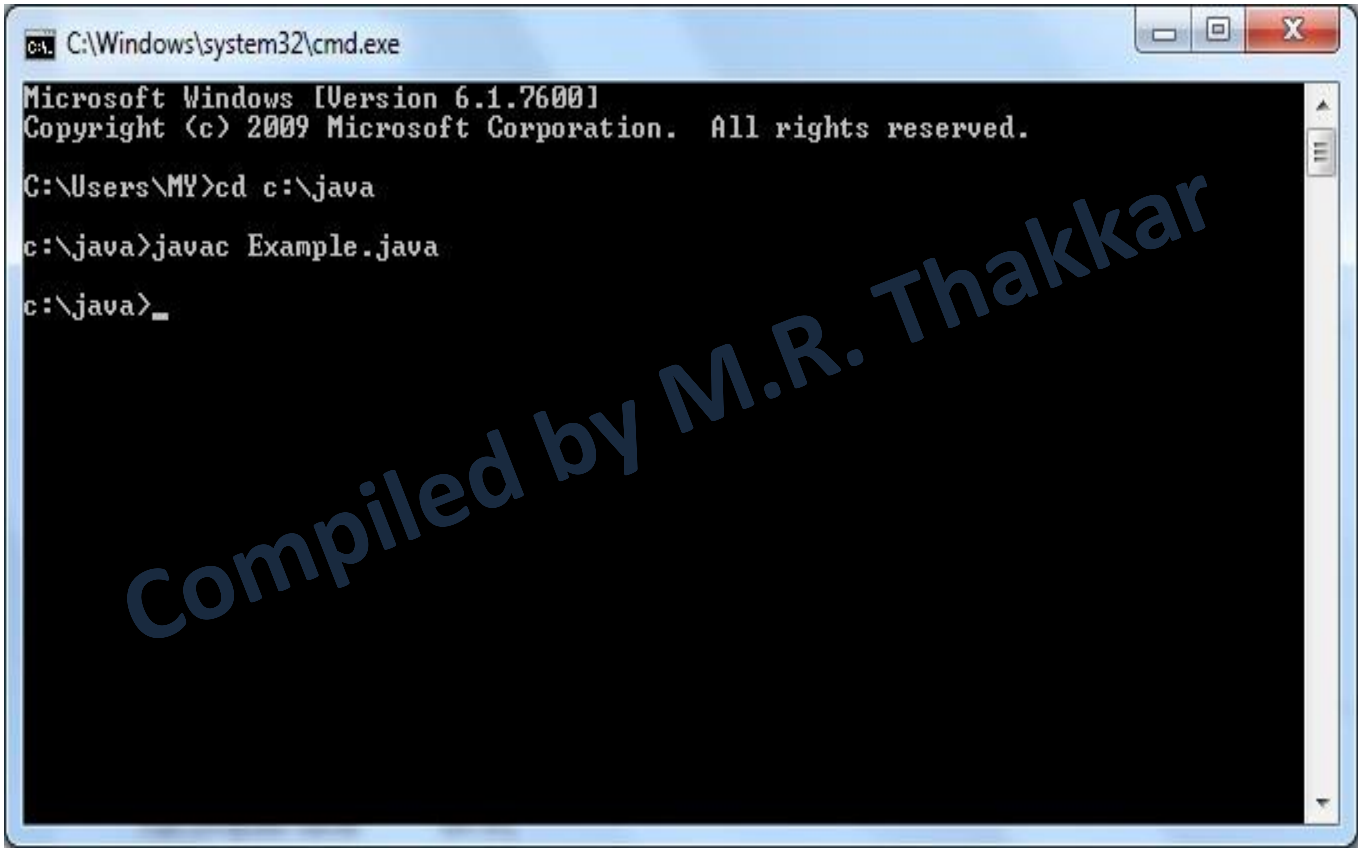
1.7 Hello World Program

- Use **javac** command to compile the file
- Enter "**javac Example.java**" command



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\MY>cd c:\java

c:\java>javac Example.java
```

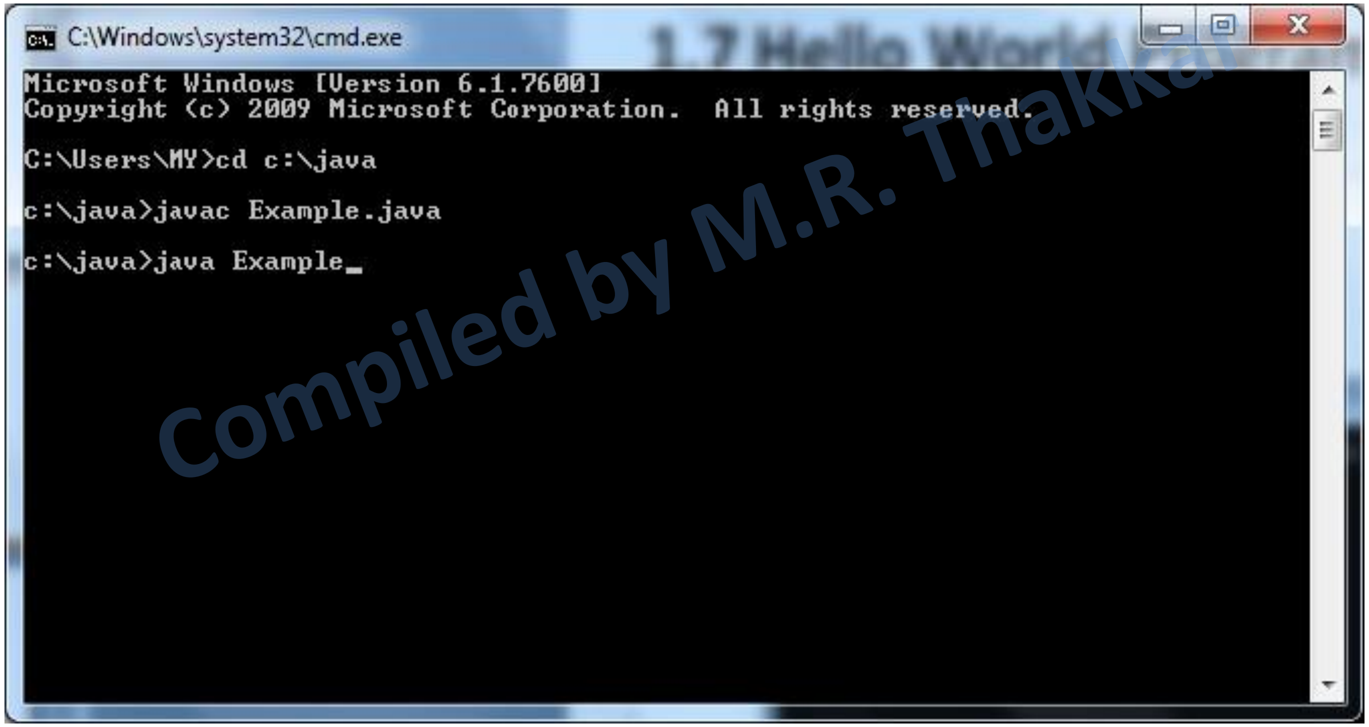
1.7 Hello World Program



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\MY>cd c:\java

c:\java>javac Example.java

c:\java>_
```

1.7 Hello World Program

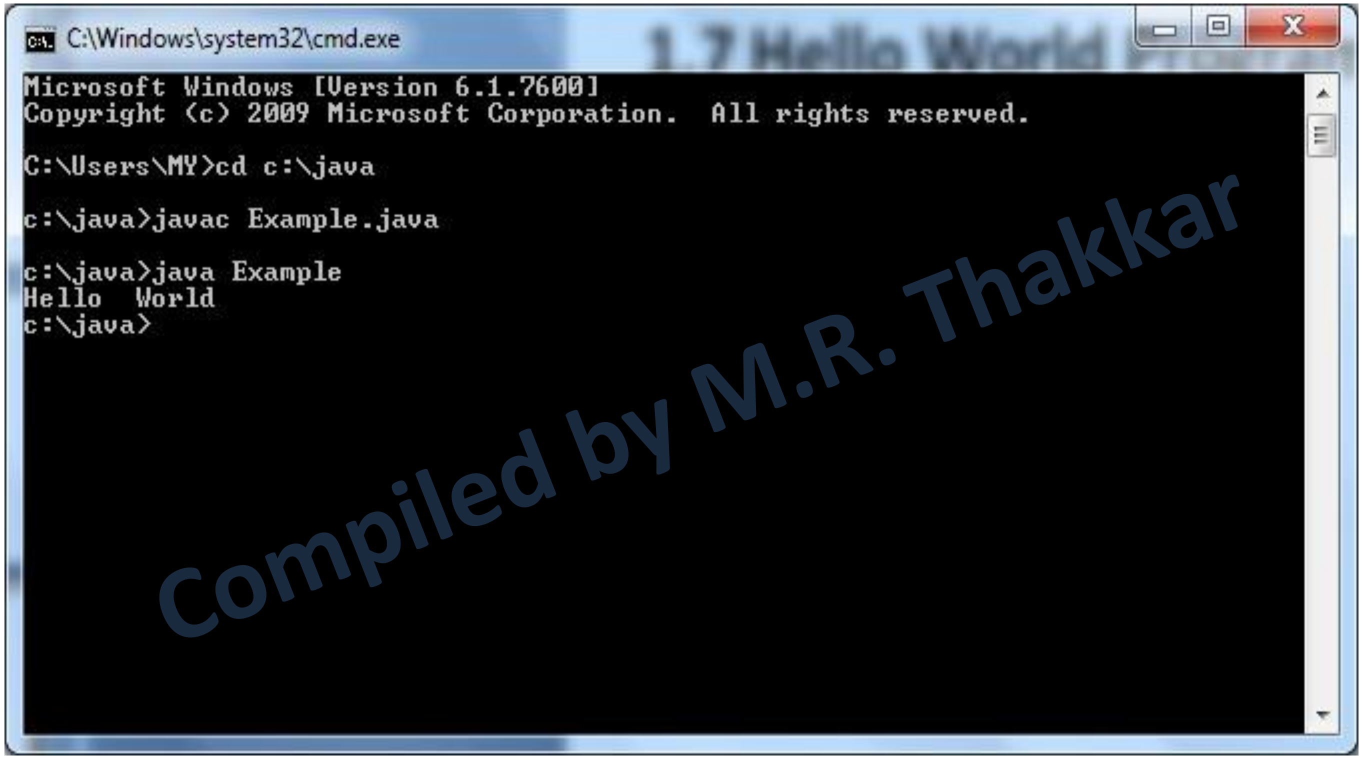
- Use java command to Run the file
- Enter "java Example" command



C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\MY>cd c:\java

c:\java>javac Example.java

c:\java>java Example_
```

1.7 Hello World Program



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\MY>cd c:\java

c:\java>javac Example.java

c:\java>java Example
Hello  World
c:\java>
```

****