

Mobile Computing
Professor Pushpedra Singh
Indraprasth Institute of Information Technology Delhi
Java Basics
Lecture 01

Hello, welcome to the course on Mobile Computing. In the course of mobile computing we will be using android operating system. And in order to develop android application you need to have a very good knowledge of java programming language. Therefore in the beginning of this course we will devote few lectures for understanding java. If you have already done java then you may choose to use this lectures to revise your concept. If you have not done java then please use this lectures to the exercises and also learn java on your own before we start with android programming.

(Refer Slide Time: 0:52)



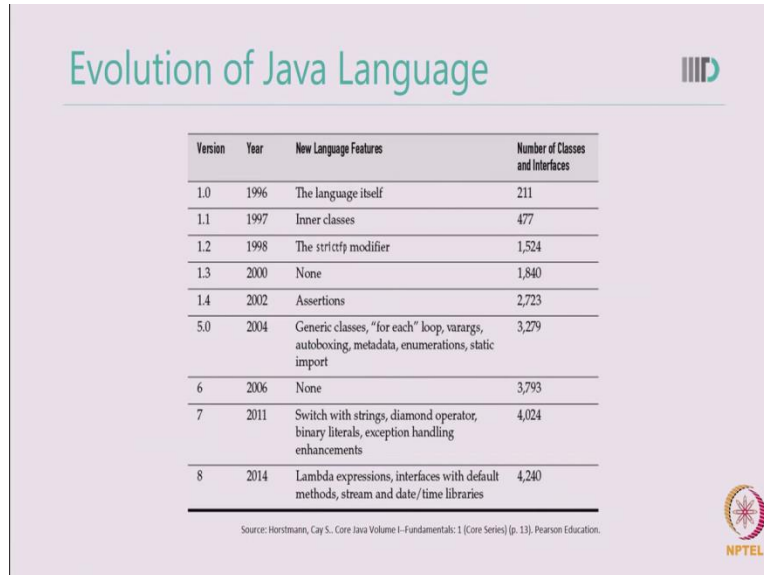
The slide is titled "Java - Origins" and features a list of four bullet points. In the top right corner, there is a logo consisting of three vertical bars of varying heights. In the bottom right corner, there is the NPTEL logo, which includes a circular emblem with a star and the text "NPTEL" below it.

- Originated in late 1995, released to public in 1996 (called Oak)
- Nine major releases since the beginning
- Current version is SDK 8.
- More than a programming language, it is a execution platform
 - Java Virtual Machine (JVM)

So, let's start with our first lecture on Java. Java originated in late 1995, and it was released to public in 1996. At that time it was called Oak. Since then there has been nine major releases current version is 8. You may download the latest Java software development kit from Oracle website and you may start programming in Java right away. More than a programming language Java is actually an execution platform you may have heard the term called the java virtual machine. Java virtual machine provides a platform to run java programs that you develop. Java virtual machine is also instrumental in making sure that once you have compiled the java


program you can run it on different types of platforms. You will learn more about java virtual machines in next lecture and also your experience of android program.

(Refer Slide Time: 1:50)



Version	Year	New Language Features	Number of Classes and Interfaces
1.0	1996	The language itself	211
1.1	1997	Inner classes	477
1.2	1998	The <code>strictfp</code> modifier	1,524
1.3	2000	None	1,840
1.4	2002	Assertions	2,723
5.0	2004	Generic classes, "for each" loop, varargs, autoboxing, metadata, enumerations, static import	3,229
6	2006	None	3,793
7	2011	Switch with strings, diamond operator, binary literals, exception handling enhancements	4,024
8	2014	Lambda expressions, interfaces with default methods, stream and date/time libraries	4,240

Source: Horstmann, Cay S. Core Java Volume I-Fundamentals:1 (Core Series) (p. 13). Pearson Education.



This is simple chart showing evolution of java language. Since starting the java language in 1996 java has grown tremendously. As you can see that the initial versions of java only have two hundred and eleven java classes. While the current version of java has around four thousand and two hundred forty java classes. This is tremendous growth and this is makes sure the java remains a very valuable programming language for today's programming scenarios.

(Refer Slide Time: 2:28)

Java Features



- As released in the “White Paper” by the authors of Java
 1. Simple
 2. Object-Oriented
 3. Distributed
 4. Robust
 5. Secure
 6. Architecture-Neutral
 7. Portable
 8. Interpreted
 9. High-Performance
 10. Multithreaded
 11. Dynamic



Java has several features I will go through the basic features that were pointed out in the white paper released at the time of release of java. Simple, object oriented, distributed, robust, secure, architecture- neutral, portable, interpreted, high performance, multi threaded and dynamic. At the time of java released we had the programming languages like C plus plus available for us. While C plus plus allowed us to use object oriented programming concepts. Java (()) (3:07). In java it is not possible to create a program that is not using object oriented programming concepts.

This makes java program easy to understand and easy to extend. Java from the very beginning also supported distributed programming by providing several libraries which enables socket programming. Java virtual machine ensures that java program can run on different architectures. Java also has a catch phrase of compile once work anywhere. Java programs are portable they can be take into different machines. The java virtual machine can interpret those programs and run set.

(Refer Slide Time: 3:51)

Java Features

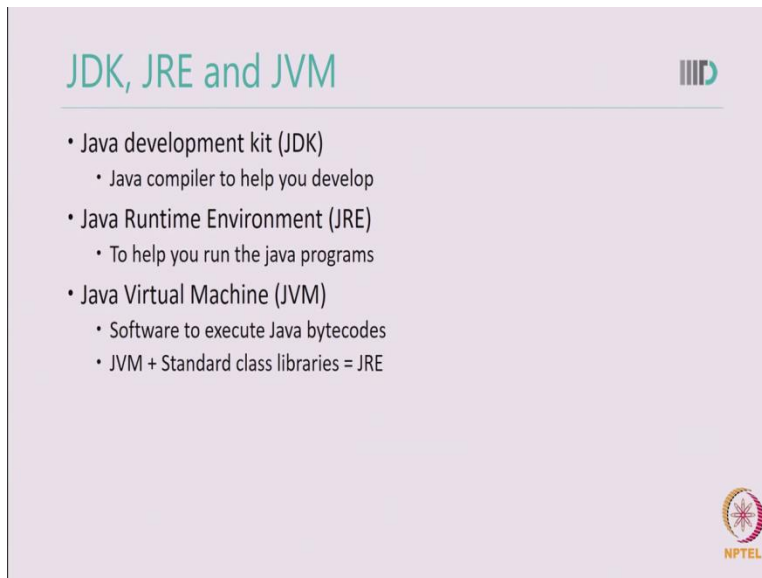


- As released in the “White Paper” by the authors of Java
 1. Simple
 2. Object-Oriented
 3. Distributed
 4. Robust
 5. Secure
 6. Architecture-Neutral
 7. Portable
 8. Interpreted
 9. High-Performance
 10. Multithreaded
 11. Dynamic



From the very beginning java supports multithreading and java programs are optimized for high performance. You will learn more about these features when you start programming android application or from your past experience programming java you may have already experience some of these features and actions.

(Refer Slide Time: 4:14)



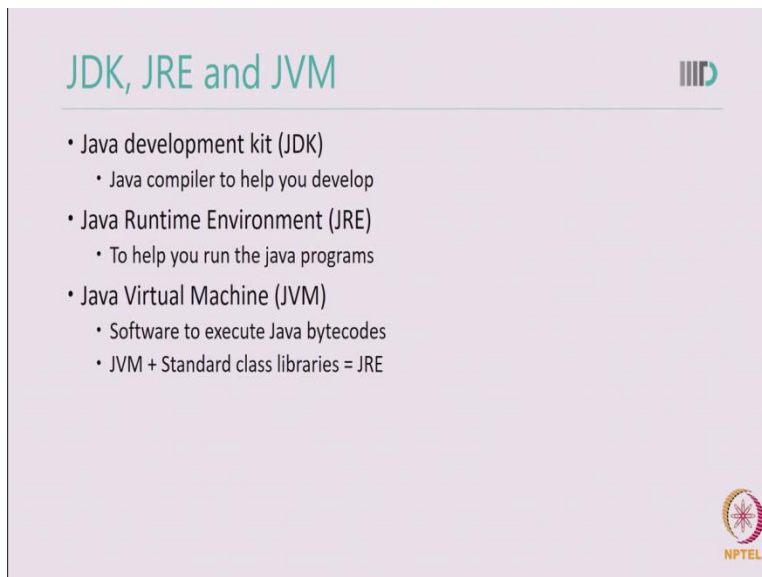
The slide features a light purple background. At the top left, the title 'JDK, JRE and JVM' is written in a teal font. To the right of the title is a small icon consisting of three vertical bars of increasing height. Below the title is a horizontal line. The main content is a bulleted list:

- Java development kit (JDK)
 - Java compiler to help you develop
- Java Runtime Environment (JRE)
 - To help you run the java programs
- Java Virtual Machine (JVM)
 - Software to execute Java bytecodes
 - JVM + Standard class libraries = JRE

In the bottom right corner, there is a circular logo with a star-like pattern and the text 'NPTEL' below it.

You may have also heard terms of JDK, JRE and JVM. JDK or SDK refers to the development kit that is available for you to write java program. JRE refers to the run time environment that is needed to run the java program. If you only want to run java programs you don't need JDK or SDK. However if you want to build your own java program you will need JDK or SDK.

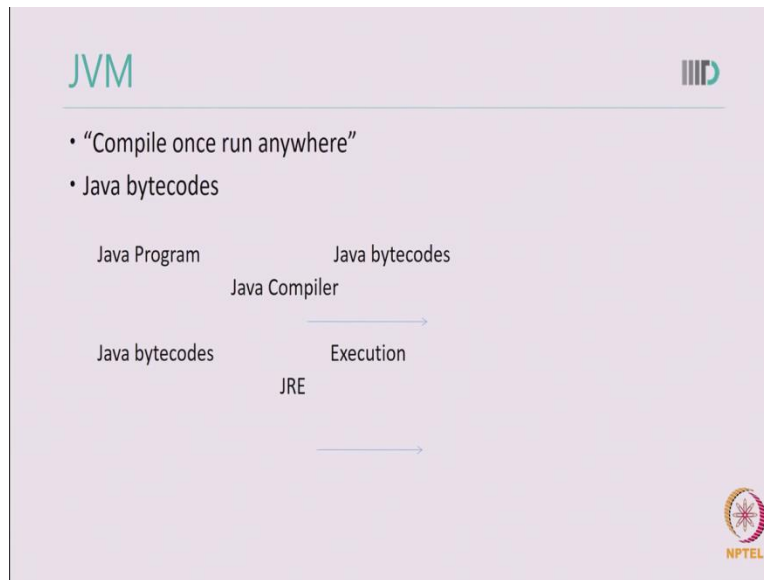
(Refer Slide Time: 4:41)



This slide is identical to the one above, featuring the same title, bulleted list, and NPTEL logo on a light purple background.

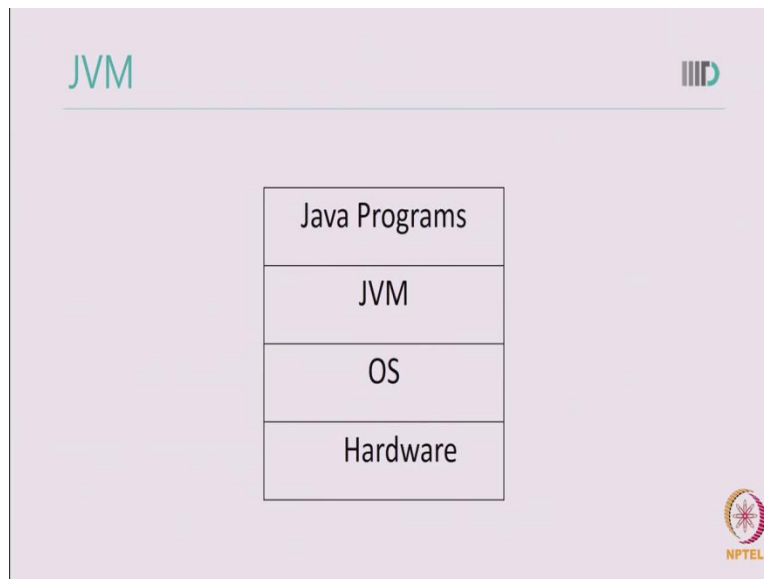
Java virtual machine is the platform that runs the java program for example when you compile a java program all you create is java byte codes. JVM takes those java bytes code and run them on a given platform.

(Refer Slide Time: 4:57)



Java has this catch phrase of compile once run anywhere. Let's see how that help in writing program that are executable on different architectures. Whenever you write a java program java compiler changes it to java byte code and then those java byte codes are ready for execution on different platforms by using JRE. As a programmer you need not to worry that on what platform your java program will run. This is unlike C plus plus where behavior of program may be different from one platform to another platform.

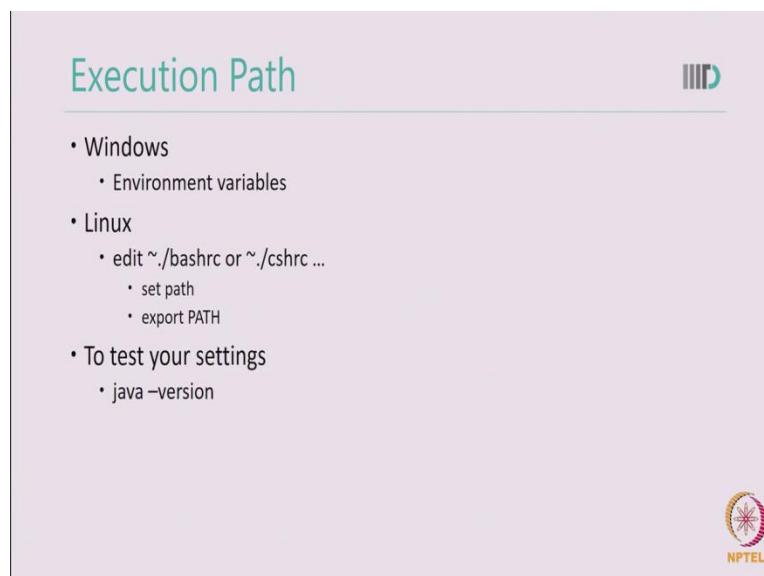
(Refer Slide Time: 5:34)



This is a simple structure showing how the java program lives on a given machine. You may have a hardware on top of the hardware you may have different OS. For example Linux, windows or mac operating system. You have a different JVM for different Operating systems. And then you have a java programs. As a developer you write a java program and then JVM hides all the complexity of the underline operating system to hardware from you.

So as a programmer you need not to worry whether your program will run on a mac machine or your program will run on Linux machine. Your program will behave exactly the same as it would from one platform to another. This is major feature of java why it was chosen to be the programming language for android operating system. As you may already know that android phones are build by different manufacturers and they come with different hardware specifications. However java makes sure those once an app has been develop and compile it behaves all most same on all these different mobile devices.

(Refer Slide Time: 6:50)

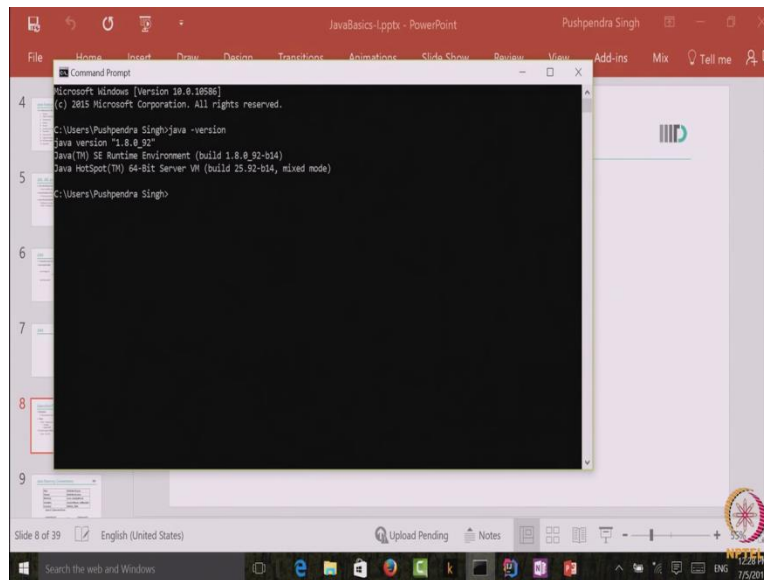


The slide, titled "Execution Path", is presented on a light purple background. It features a list of instructions for setting up Java on different operating systems. The list is organized into three main bullet points: "Windows", "Linux", and "To test your settings". Under "Windows", there is a sub-bullet "Environment variables". Under "Linux", there are sub-bullets for editing shell files and setting environment variables. The "NPTEL" logo is visible in the bottom right corner of the slide.

- Windows
 - Environment variables
- Linux
 - edit ~/.bashrc or ~/.cshrc ...
 - set path
 - export PATH
- To test your settings
 - java -version

After you have downloaded java on your platform you will have to Set the path correctly so that your system can recognize java program in windows operating (sys) in windows operating system you do it using environment variable in Linux you do it by bashrc or cshrc files or other shell files that you may be using. You can test your setting by running java – version command.

(Refer Slide Time: 7:17)



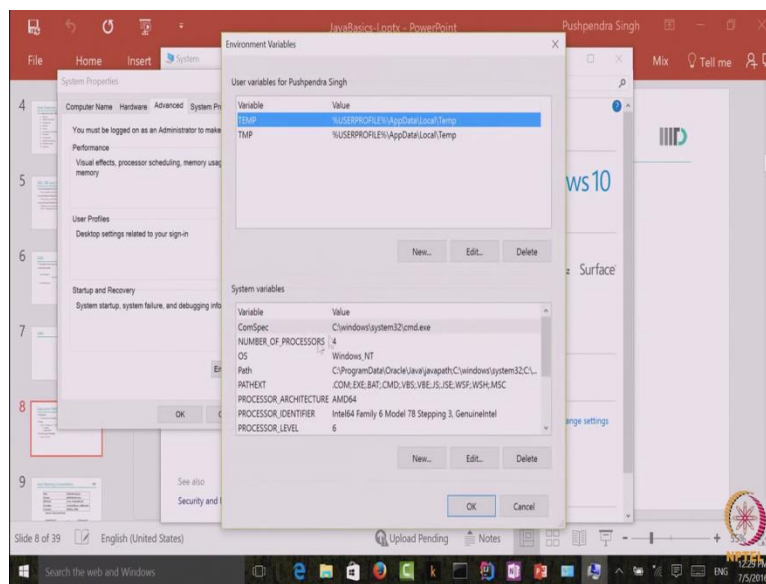
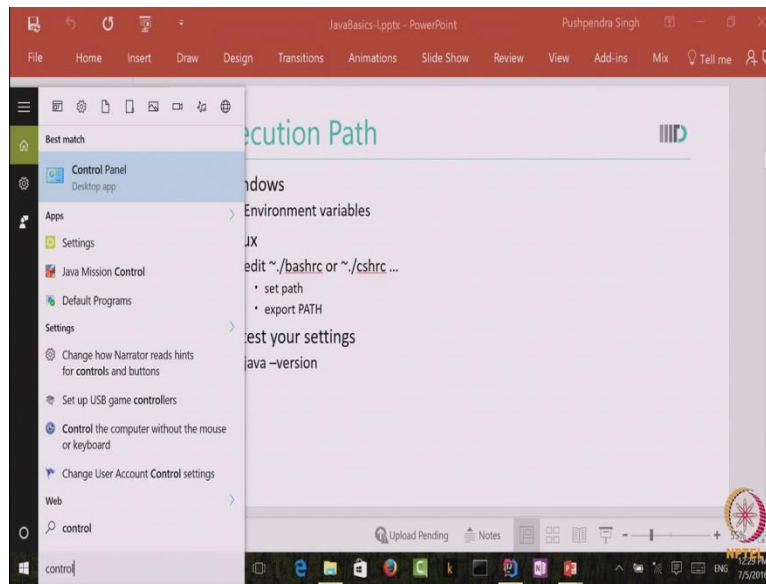
```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

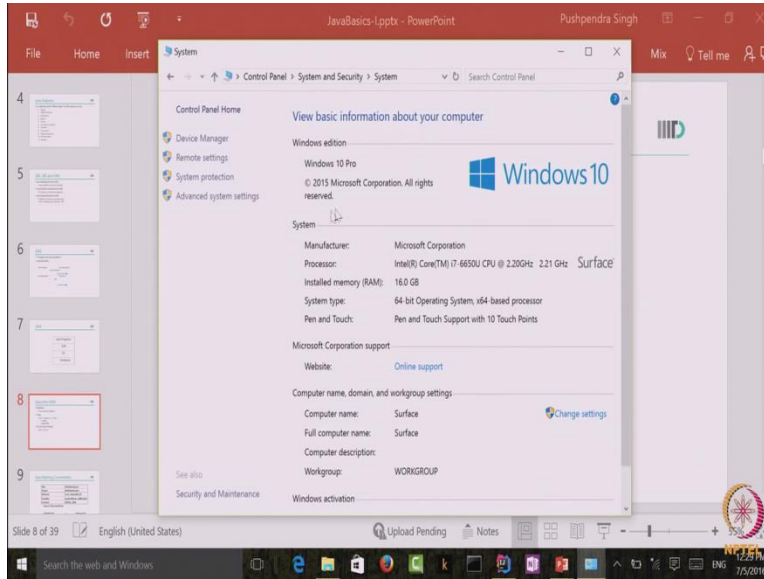
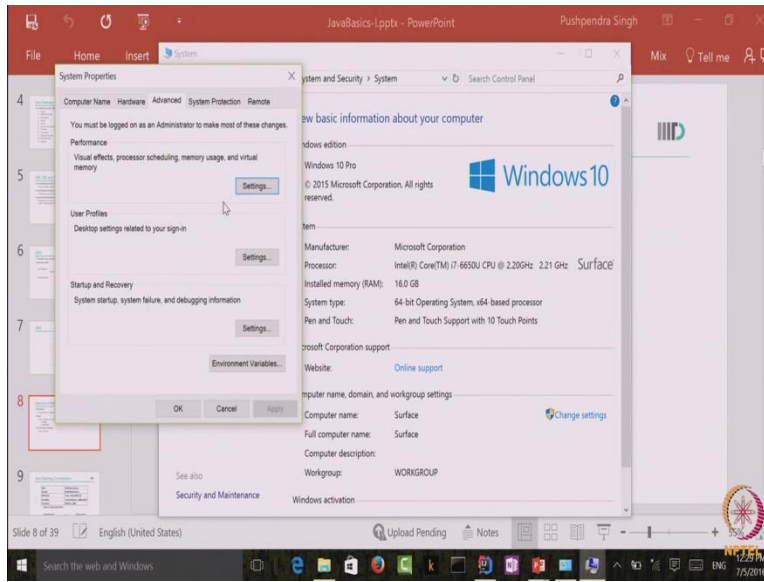
C:\Users\Pushpendra Singh>java -version
java version "1.8.0_92"
Java(TM) SE Runtime Environment (build 1.8.0_92-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.92-b14, mixed mode)

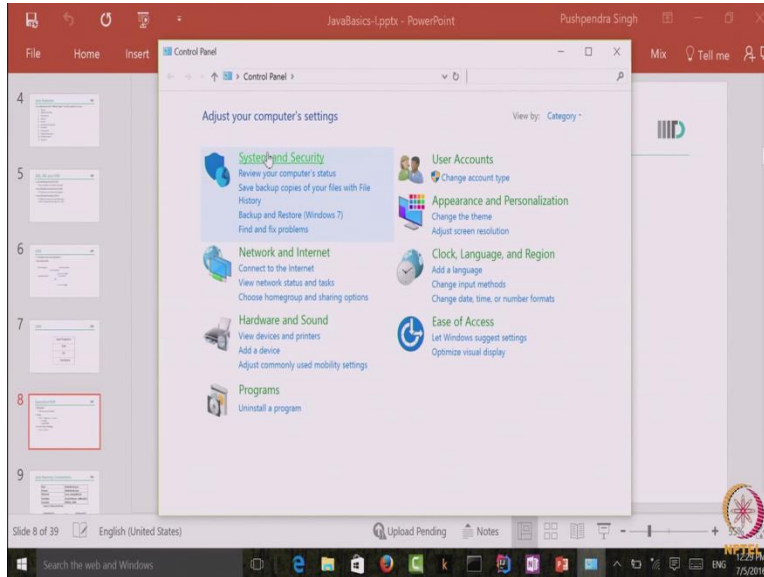
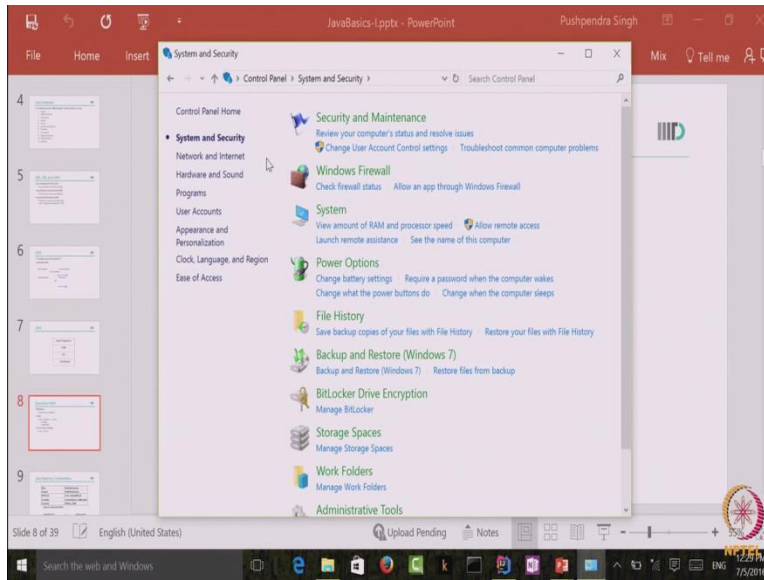
C:\Users\Pushpendra Singh>
```

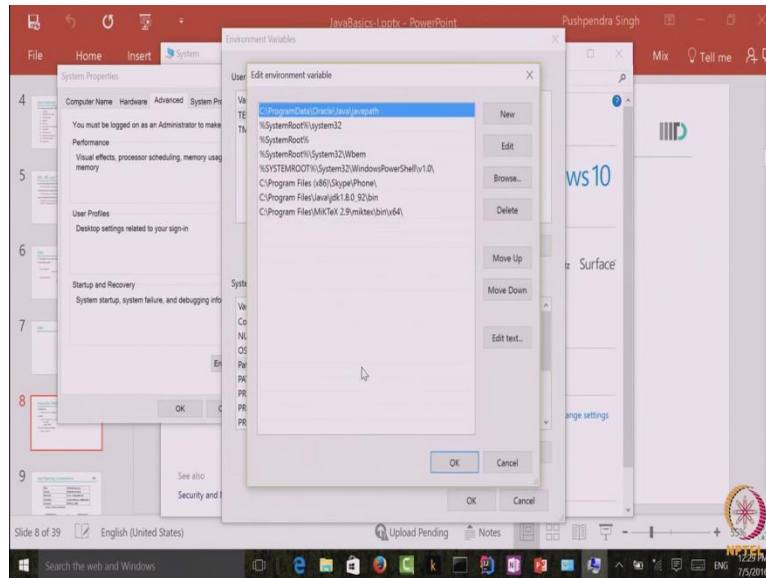
Let me show you a quick example, this is the command prompt of a windows machine. If I type java version it displays the latest java SDK that is installed on my machine. You may find it difficult to read, so I will read it for you. Currently it says java version 1.8.0 underscore 92 that means that I have SDK 8 installed and with the update 92 which is the latest update. You may get a different value depending on the SDK that you have installed. However I advised you to install latest SDK the current SDK release is 8 and therefore if your java- version showing any value less then please update it to the latest SDK. You will download java SDK from the oracle website.

(Refer Slide Time: 8:09)



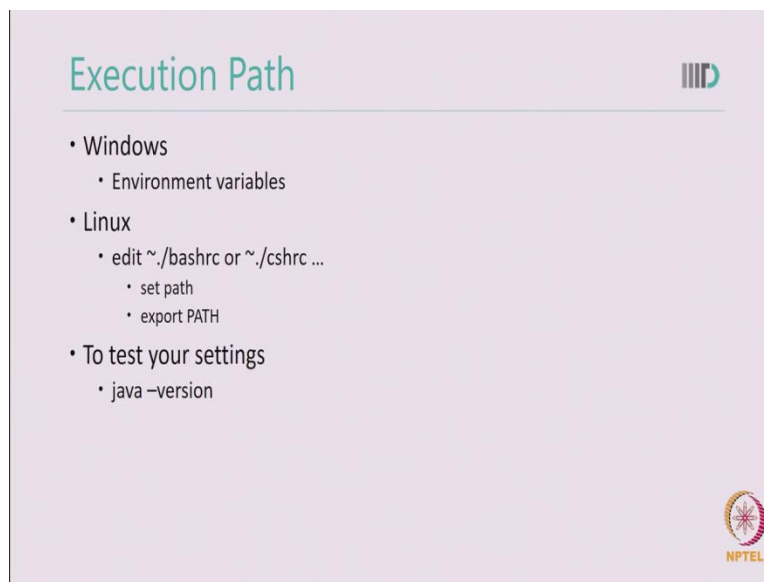






Once you have downloaded you may set those values in the control panel. We choose to go to systems. Go to advance system setting. Go to environment variables. Take your path, and make sure that your java library is available. So (he) here you can see that my JDK directory is available in the path.

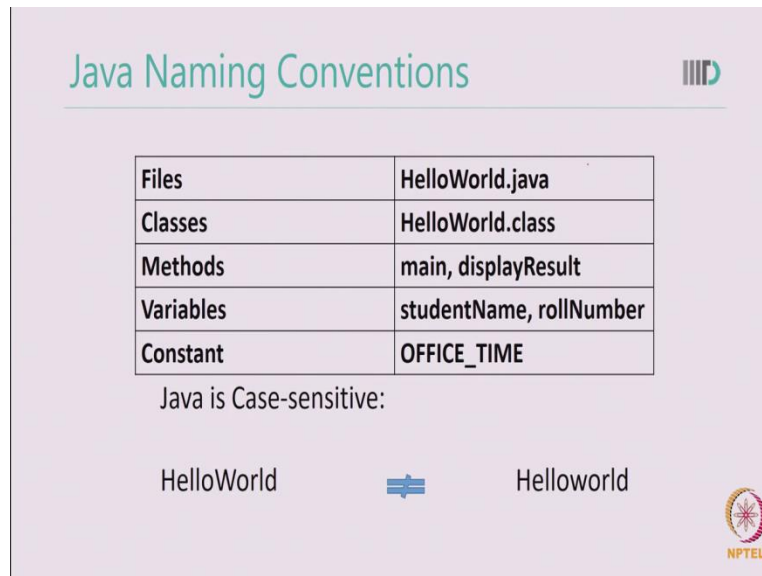
(Refer Slide Time: 8:37)



Once you have installed java, run the java – version command and see that your program can recognize where your java is installed. Once you have installed java, run java – version on your shell and check the value. If it displays the latest SDK that you have installed then you have correctly installed java. If it does not then try to check your installation. There are several

websites available which tells you how to install a Java including the website on a Oracle which currently owns java. I hope you have successfully installed java now let's go and try to understand what java programming language is.

(Refer Slide Time: 9:23)



The slide is titled "Java Naming Conventions" and features a table with the following content:

Files	HelloWorld.java
Classes	HelloWorld.class
Methods	main, displayResult
Variables	studentName, rollNumber
Constant	OFFICE_TIME

Below the table, it states "Java is Case-sensitive:" and shows "HelloWorld" and "Helloworld" with a blue double-line arrow pointing from the capital 'H' in the first word to the lowercase 'h' in the second word, indicating they are not equivalent.

Lets first start with the java naming conventions. Java is a case sensitive language which means that small and capital letters even though they may refers to the same word means different for java programming language. In java hello word written in a capital H and capital W is different then hello word written with only one capital word. Java uses naming conventions which make sure that programmer other than you can understand your program easily. While (9:53) in java we named each java class with a capital letter starting and every time (we)when we use a different word we use a capital letter.


So if I have to declare a java class by hello world I will use H as a capital and W as a capital. All the methods in java start with the small letter. But if it is a combination of different word then the second letter starts with a capital letter. So a main is all a small while display result starts with a small but then the result is a capital R. Variables follow the same naming conventions as method and constant use all capital. If you don't follow the naming conventions the java compiler will not stop. However other java programmers may not find your program very readable. I advised you to go through the detail java naming convention which are given on a oracle website under java documentation and use the naming conventions for your program.


(Refer Slide Time: 11:02)

Java Naming Conventions

Files	HelloWorld.java
Classes	HelloWorld.class
Methods	main, displayResult
Variables	studentName, rollNumber
Constant	OFFICE_TIME

Java is Case-sensitive:

HelloWorld  Helloworld



Let a see a simple java program that you may run before running java apart from JDK I advised you to also install an IDE. IDE stands for Integrated Development Environment. There are different java IDE which are available the popular IDE is intelliJ another popular IDE is Eclipse or Netbeans. (in) For this program we will be using intelliJ IDE which is also a base IDE for android studio that we will be using later for developing android applications. I have already installed intelliJ on my system. You may find intelliJ on internet available for download.

(Refer Slide Time: 11:46)

```
package in.ac.iiitd.nptel;  
  
public class Main {  
  
    public static void main(String[] args) {  
        // write your code here  
        System.out.println("Hello World!");  
    }  
}
```

"C:\Program Files\Java\jdk1.8.0_92\bin\java" ...
Hello World!
Process finished with exit code 0

Compilation completed successfully in 800ms (moments ago)

```
package in.ac.iiitd.nptel;  
  
public class Main {  
  
    public static void main(String[] args) {  
        // write your code here  
        System.out.println("Hello World!");  
    }  
}
```

"C:\Program Files\Java\jdk1.8.0_92\bin\java" ...
Hello World!
Process finished with exit code 0

Run selected configuration

```
package in.ac.iiitd.nptel;  
  
public class Main {  
  
    public static void main(String[] args) {  
        // write your code here  
    }  
}
```

"C:\Program Files\Java\jdk1.8.0_92\bin\java" ...
Process finished with exit code 0

Compilation completed successfully in 2s 179ms (moments ago)

```
package in.ac.iiitd.nptel;  
  
public class Main {  
  
    public static void main(String[] args) {  
        // write your code here  
    }  
}
```



```
FirstDemo - [C:\Users\Pushpendra Singh\IdeaProjects\FirstDemo] - Main.java - IntelliJ IDEA 2016.1.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FirstDemo
Main.java
package in.ac.iitd.nptel;

public class Main {

    public static void main(String[] args) {
        // write your code here
    }
}

Scanning files to index
Search the web and Windows
CTRL+LTT:RT
NPTEL
12:31 PM
7/5/2016
```

New Project

Project name:

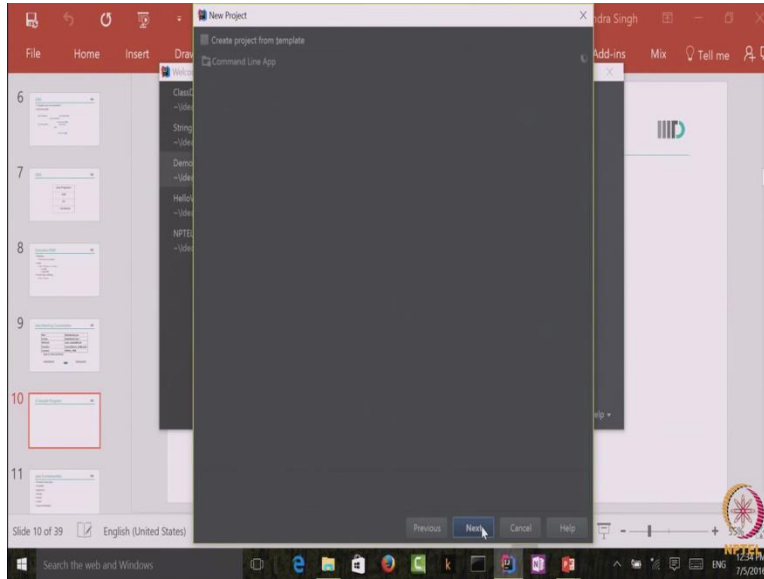
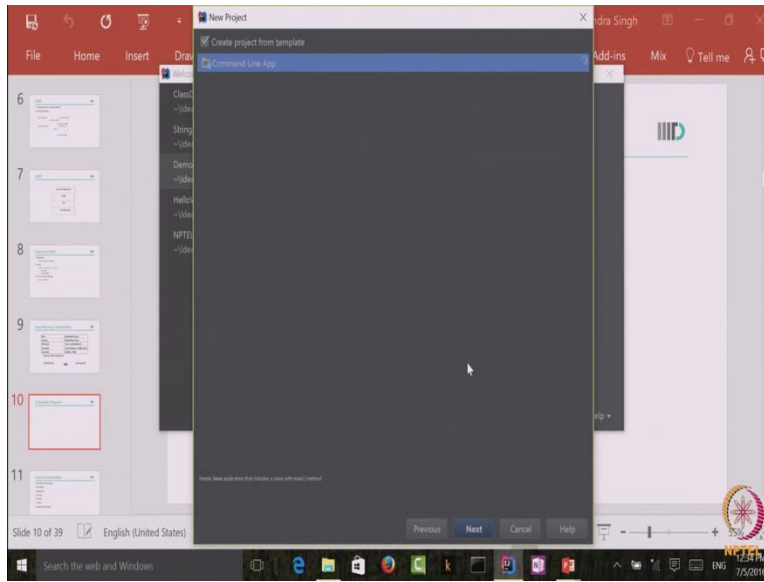
Project location: C:\Users\Pushpendra Singh\IdeaProjects\untitled

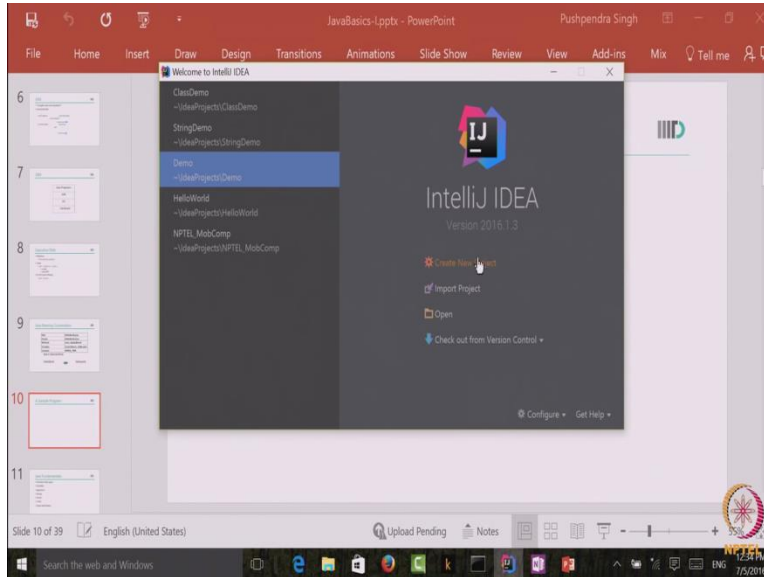
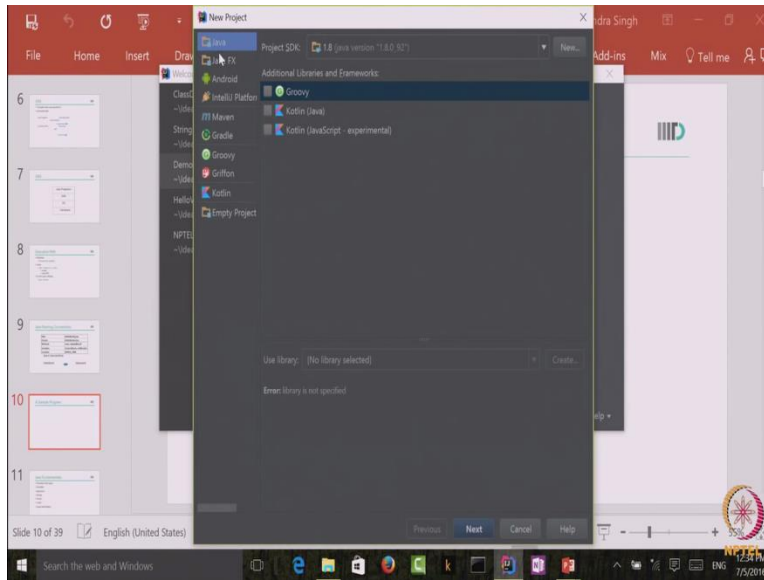
Base package: in.ac.iitd.nptel

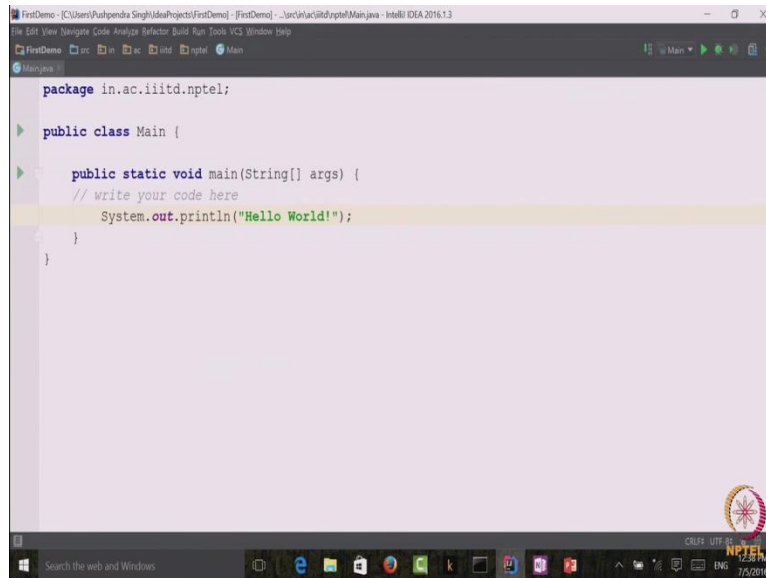
- Class
- Idea
- String
- Idea
- Demo
- Idea
- Hello
- Idea
- NPTEL
- Idea

Slide 10 of 39 English (United States) Previous **Finish** Cancel Help

Search the web and Windows
NPTEL
12:31 PM
7/5/2016





The image shows a screenshot of the IntelliJ IDEA 2016.1.3 IDE. The main editor window displays a Java source file for a class named 'Main' in the package 'in.ac.iiitd.nptel'. The code is as follows:

```
package in.ac.iiitd.nptel;

public class Main {

    public static void main(String[] args) {
        // write your code here
        System.out.println("Hello World!");
    }
}
```

The IDE interface includes a menu bar at the top with options like 'File', 'Edit', 'View', 'Navigate', 'Code', 'Analyze', 'Refactor', 'Build', 'Run', 'Tools', 'VCS', 'Window', and 'Help'. A toolbar below the menu bar contains icons for various actions. The Windows taskbar is visible at the bottom, showing the search bar and several application icons. The system tray in the bottom right corner displays the date and time as '7/5/2016 12:58 PM'.

This is the IntelliJ IDE which allows me to program Java and also run Java programs. Let's create a new project. As you see that IntelliJ gives me a choice of creating different types of projects; we want to create only a Java project for the time being. I have chosen the SDK that is installed; I would like to create a project from a template. A template only makes sure that basic functionality is there; I will need to write all the extra functionality that I need for my program. I give a name, let's give a name 'First Demo'. You will see the base package I have defined as 'in.ac.iiitd.nptel'. The package makes sure that your Java programs are stored in a corresponding directory structure.

Now IntelliJ has created my basic program. This program has nothing but a simple main class with a single main function. I can run this program, but as you can see, the program doesn't do anything except compiling and finishing. Now let me write 'Hello World' here so that we can see 'Hello World' being displayed. To display anything in Java, we use the method called 'System.out.println'. In Java, we end every line with a semicolon.

Now your Java program is ready to display 'Hello World'. And here you see, so you have created your first 'Hello World' program in Java using IntelliJ IDE. If you want to use another IDE such as Eclipse or NetBeans, that is fine. However, for this program, we will only support IntelliJ IDE. Congratulations, you have built your first 'Hello World' program in Java. We end this lecture now. In the next lecture, we will learn about Java fundamentals. Thank you!