|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **SUKKUR IBA UNIVERSITY**  MERIT – QUALITY – EXCELLENCE | | | |  |
| Lab handout 01 | | | | | |
| Object Oriented Programming (CSE-211), Spring 2021 | | | | | |
| Name: **Hasnain Ali** | | CMS ID#: **033-19-0028** | | Instructor: Dr. Abdul Aziz | |
| Section: A Lab group: **X** | | Department: Electrical Engineering | | Marks obtained: out of 100% | |
| **NOTE: Submit before next lab** | | | **Submitted on 04-02-2021** | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RUBRIC | | | | | |
| **Criteria** | **Fully meets criteria (80~100 %)** | **Almost meets criteria (60~79%)** | **Progressing to meet criteria (40~59%)** | **Below Expectations (0~39%)** | **Total out of 100%** |
| **Arrival time, attention, discipline, behavior, willingness, participation** | Excellent | Good | Average | Poor | **80%** |
| **Understanding of programming, concepts, creativity** | Excellent | Good | Average | poor | **71%** |
| **Computer, Java IDE, editor proficiency** | Excellent | Good | Average | poor | **52%** |
| **Each program prints your name as a first line** | Excellent | Good | Average | poor | **99%** |
| **Equipment handling** | Excellent | Good | Average | poor | **83%** |
| **Guidance required** | No | Little | yes | Lot of guidance required | **39%** |
| **Helping group/class mates/teacher** | Extremely helpful | Helpful | Don't do or ask for help | Require help | **70%** |
| **Coding layout** | Excellent | Good | Average | poor | **78%** |
| **Readability of screenshots** | Excellent | Good | Average | poor | **39%** |
| **Program results** | Perfect | Acceptable | Minor variations | Different than required | **79%** |
| **Methodology** | Planned task extremely well and followed instructions | Planned task well and followed instructions | Planned and followed task instructions roughly | Did not plan well and did not follow instructions fully | **55%** |
| **Mentioned program number and title** | Excellent | Good | Average | poor | **100%** |
| **Before leaving lab, student placed everything back at proper place, turned off equipment etc.** | Excellent | Good | Average | poor | **30%** |
| **Submitted handout on time** | Yes | Less than 1 day late | Less than 3 days late | A week or later | **100%** |
| **Followed guidelines to solve and submit handout without cheating** | Excellent | Good | Average | poor | **100%** |
| **Minimum score in any of the criteria** | | | | | **30%** |

**Downloading, installing & setting up JDK & NetBeans IDE to write, compile, execute and debug Java programs**

**Objectives**

* **Download, install and setup JDK**
* **Download, install and setup NetBeans**
* **Learn to write, compile, execute simple Java programs using notepad**
* **Learn to write, run simple Java programs using NetBeans**
* **Differentiating and debugging programming errors**

**Hardware and software required**

PC with:

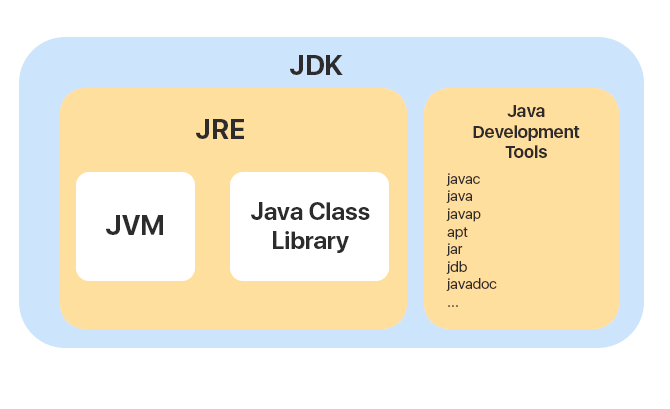
* **JDK 15.0.2**
* **NetBeans IDE 12.2**
* **Notepad**
* **Access to internet**
* **Access to LMS**
* **Online Java compilers**

**Theory**

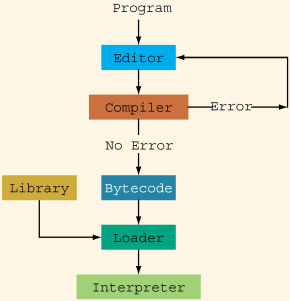
JDK stands for Java development kit and it is a software/tool for programmers who want to write and run java programs. JDK contains Java development tools such as java, javac, Javadoc etc and JRE. Java development tools are used for different purposes such as compiling (generating bytecode for JRE), debugging etc. Command “javac” is used to compile java program. Command “javadoc” is used to generate file for program description. Command “jar” is used to combine multiple classes etc to make it as one package for distribution.

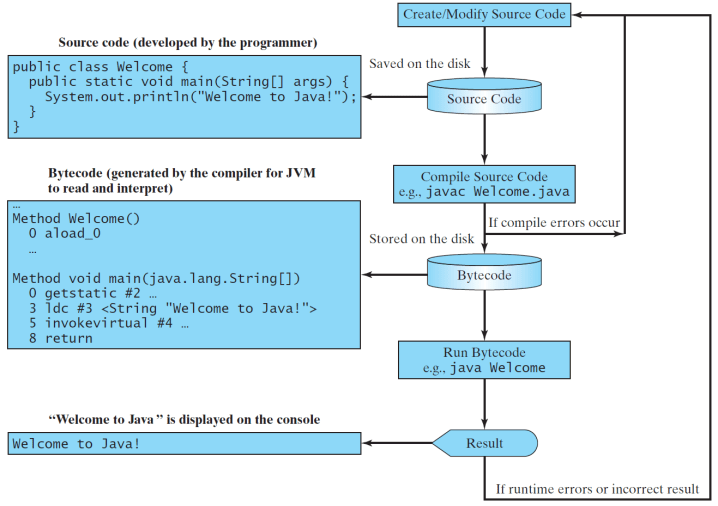
If someone don’t want develop java programs and just run java programs then they need only JRE to interpret bytecode and generate output with the help of JVM and java libraries etc.

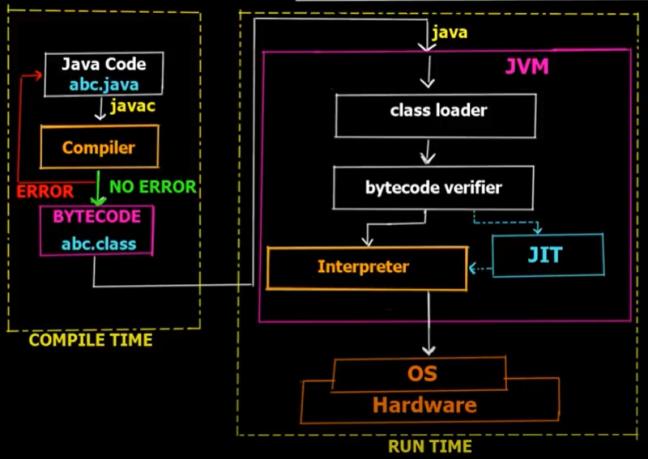
JRE which stands for Java Runtime Environment has a purpose to provide environment for java codes to be executed. Since, java byte codes are machine independent therefore there is JVM which stands for Java Virtual Machine and it is the part of JRE. JVM loads bytecode, verifies it and interprets bytecode into machine code for execution. JRE also contains libraries required by JVM for bytecode to be executed. JDK, JRE, JVM all are OS/machine dependent but not the bytecode.



Java program is run in the following steps.



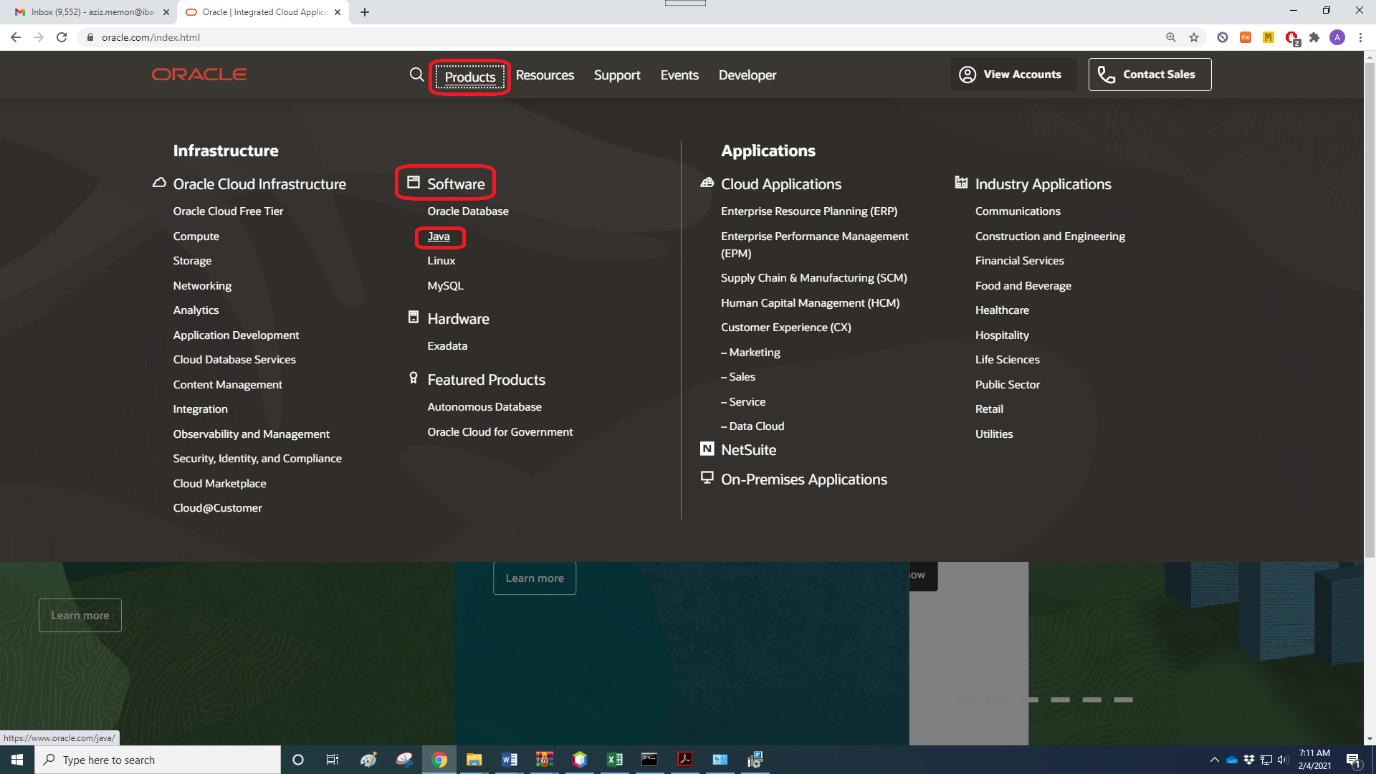




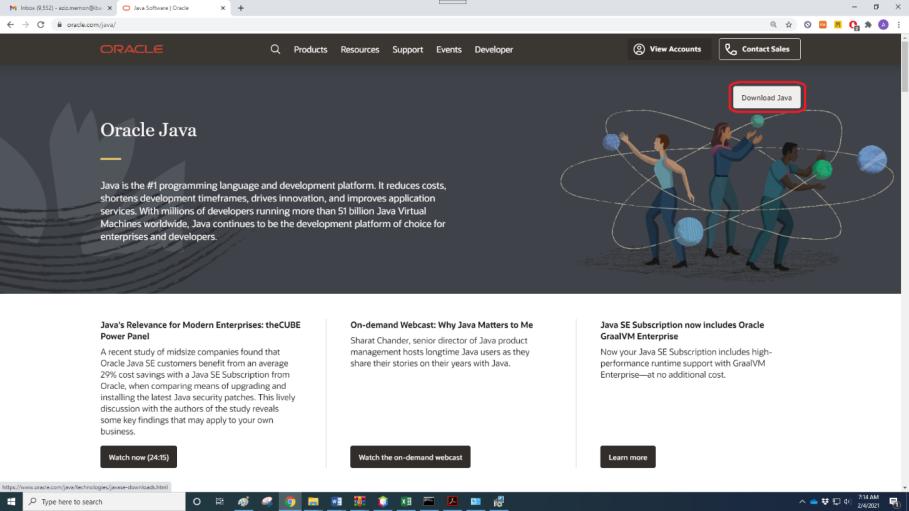
* **Download, install and setup JDK**

Browse <https://www.oracle.com/>

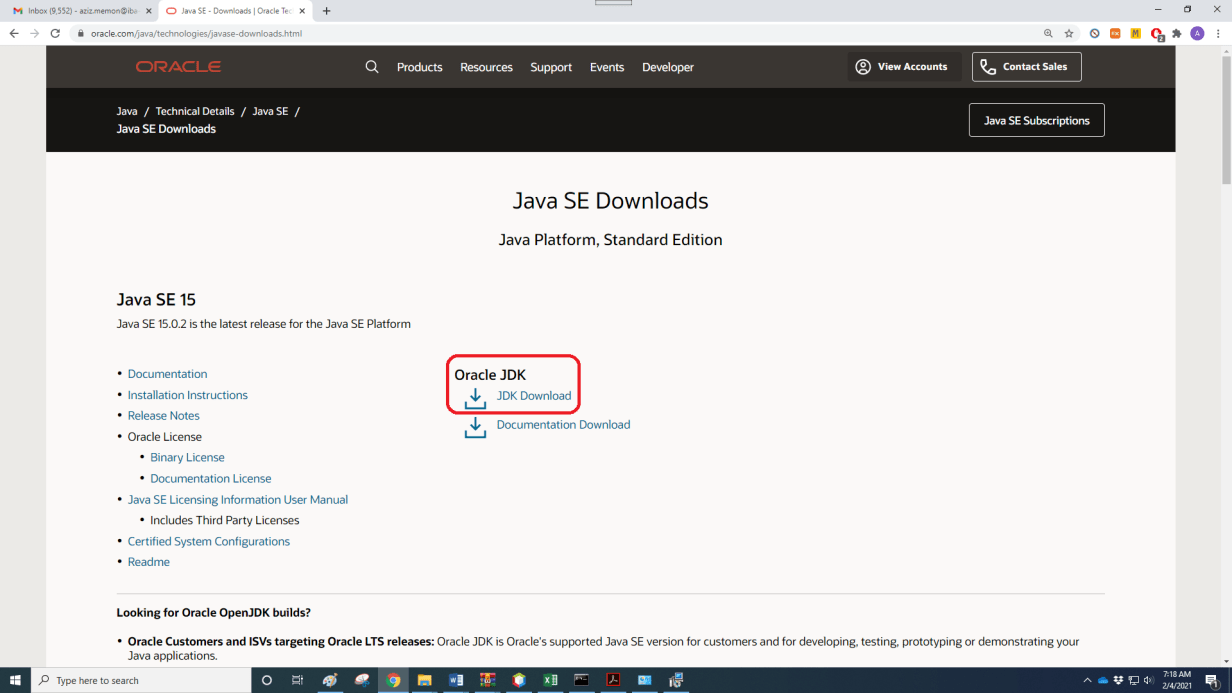
Navigate to products>Software>Java



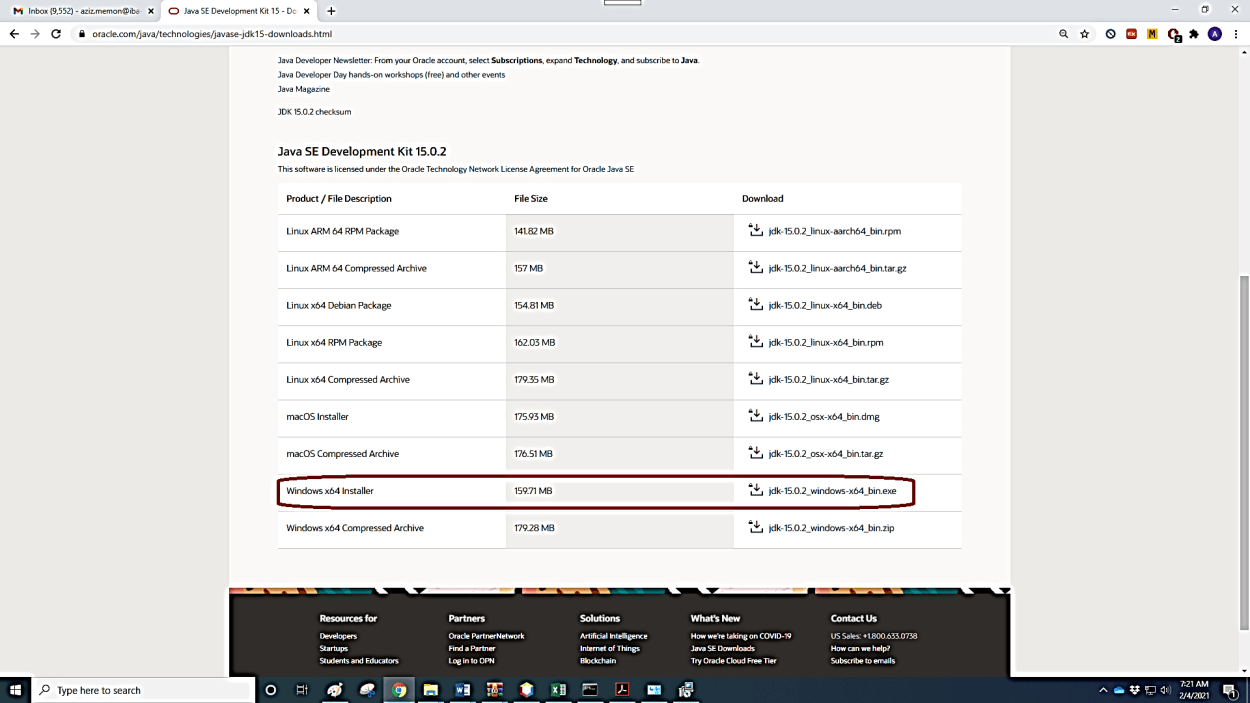
Click on download Java



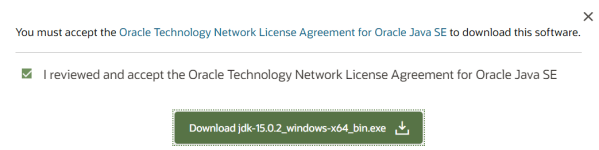
Click on JDK Download



Click on [jdk-15.0.2\_windows-x64\_bin.exe](https://www.oracle.com/java/technologies/javase-jdk15-downloads.html#license-lightbox)



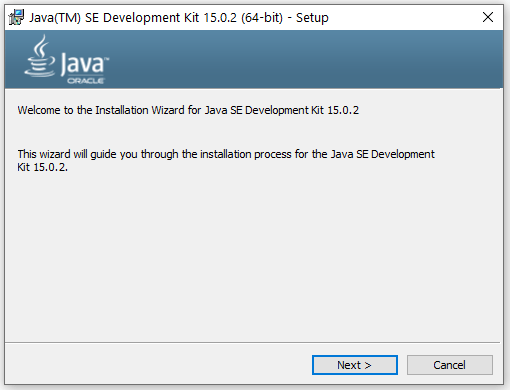
Accept agreement to stat downloading



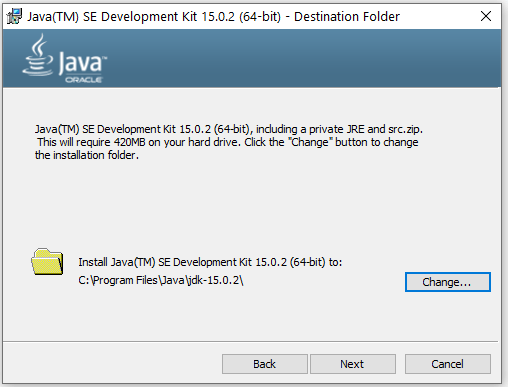
JDK can be directly downloaded by clicking following link.

<https://download.oracle.com/otn-pub/java/jdk/15.0.2%2B7/0d1cfde4252546c6931946de8db48ee2/jdk-15.0.2_windows-x64_bin.exe>

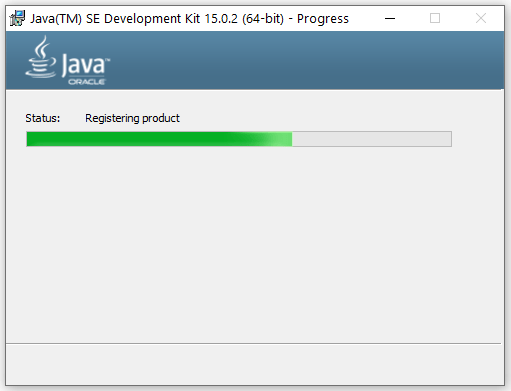
Double click downloaded JDK and click YES to allow JDK to make changes to system and then click NEXT.



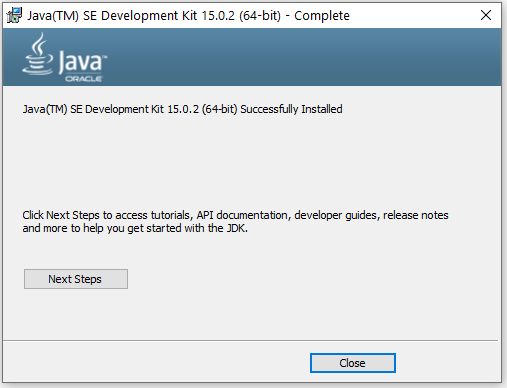
Installation location by default C:\Program Files\Java\jdk-15.0.2\



Installation in progress

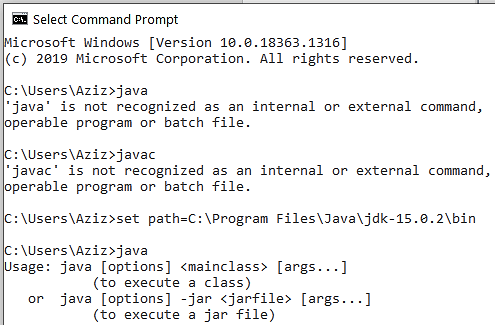


JDK successfully installed, click CLOSE



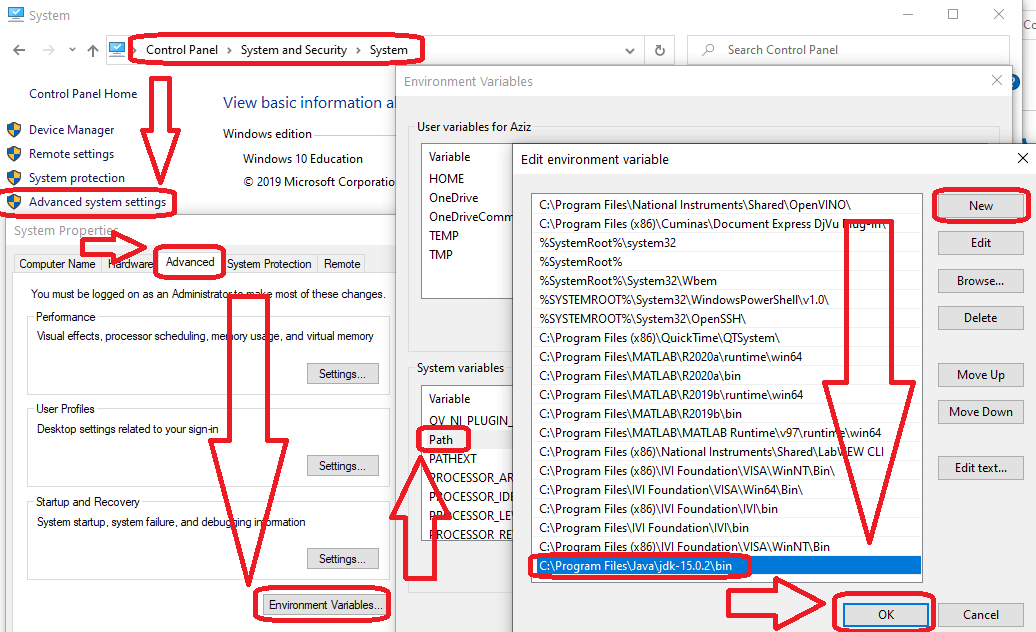
We need to set Java path so that whenever java program will be compiled or executed it can locate JDK easily. Path can be set in two ways i.e. temporarily and permanently.

To set java path temporarily, open command prompt and run command “set path=C:\Program Files\Java\jdk-15.0.2\bin”. Once the command prompt is closed, java path needs to be set again.

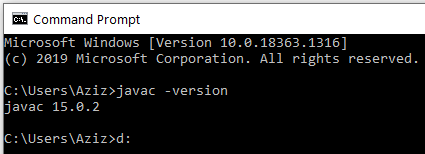


To set java path permanently, this path “C:\Program Files\Java\jdk-15.0.2\bin” should be added as per following steps.

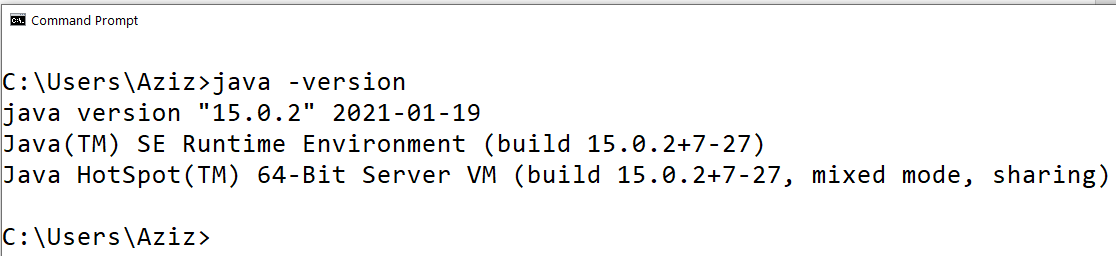
Go to Control Panel > System and Security > System, and then click Advanced system settings to view System properties window. In system properties window press advanced tab and click on Environment variables to add new path. Finally click ok and you are all set.



To check if the Java compiler is installed, open command prompt and run command “javac -version”. If the command returns something such as “javac 15.0.2”, it means JDK having Java compiler is installed successfully.



To check version information of JDK, JRE and JVM, open command prompt and run command “java -version”.

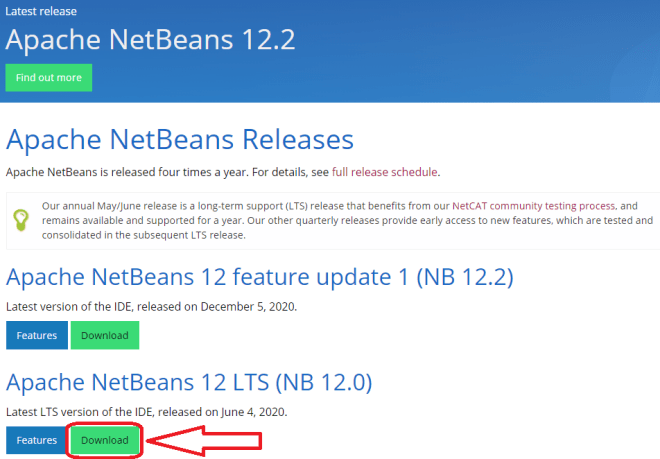


* **Download, install and setup NetBeans**

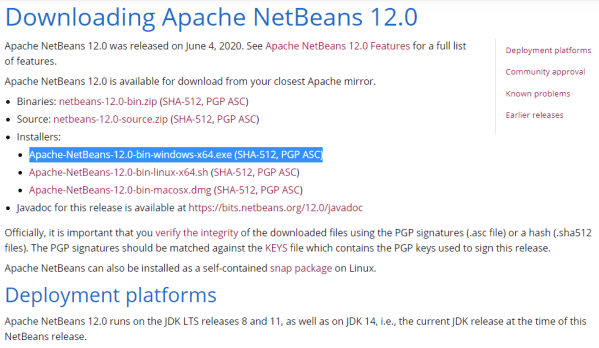
Go to website <https://netbeans.org/> and click download.



Click to download NetBeans 12.0 LTS (Long term support)



Click to download [Apache-NetBeans-12.0-bin-windows-x64.exe](https://www.apache.org/dyn/closer.cgi/netbeans/netbeans/12.0/Apache-NetBeans-12.0-bin-windows-x64.exe) ([SHA-512](https://downloads.apache.org/netbeans/netbeans/12.0/Apache-NetBeans-12.0-bin-windows-x64.exe.sha512), [PGP ASC](https://downloads.apache.org/netbeans/netbeans/12.0/Apache-NetBeans-12.0-bin-windows-x64.exe.asc))

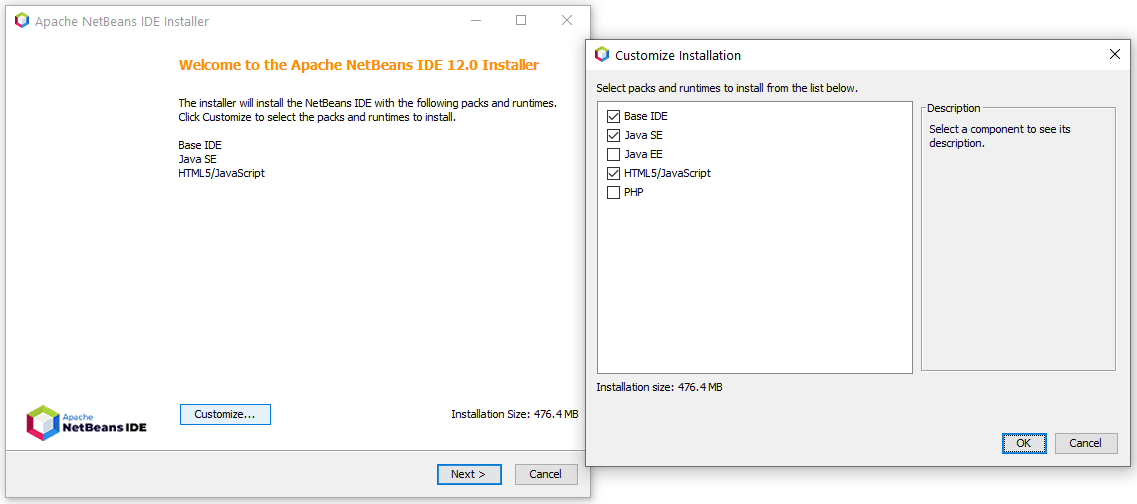


Click to download <https://downloads.apache.org/netbeans/netbeans/12.0/Apache-NetBeans-12.0-bin-windows-x64.exe>

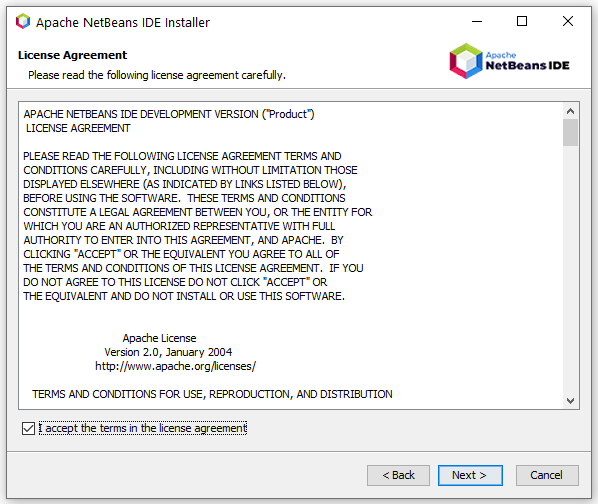


Double click on downloaded Apache-NetBeans-12.0-bin-windows-x64.exe file. Allow this application to make changes to your device.

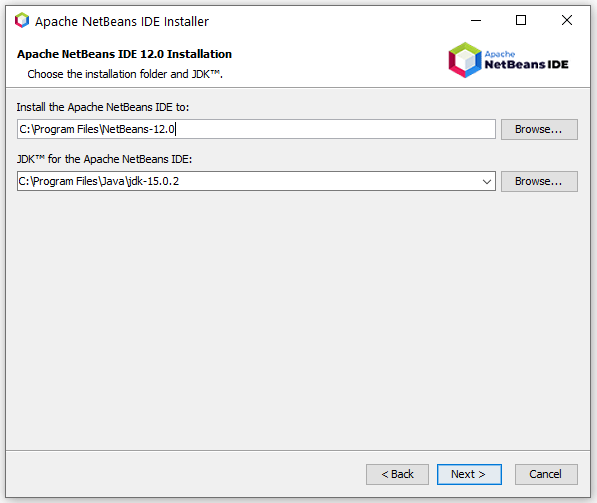
You may customize, not to install PHP and Java EE. Click next.



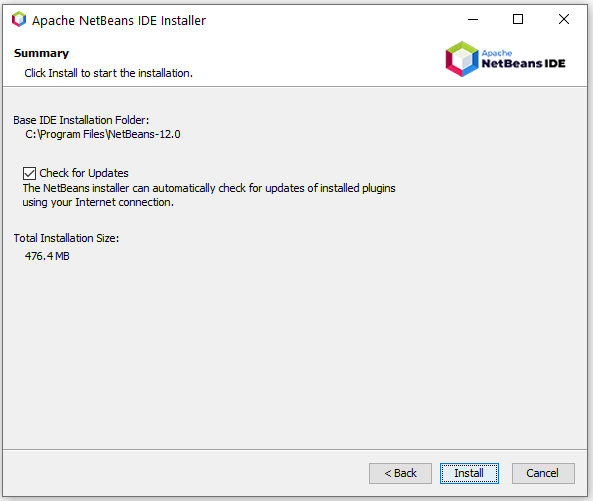
Accept license agreement and click next.



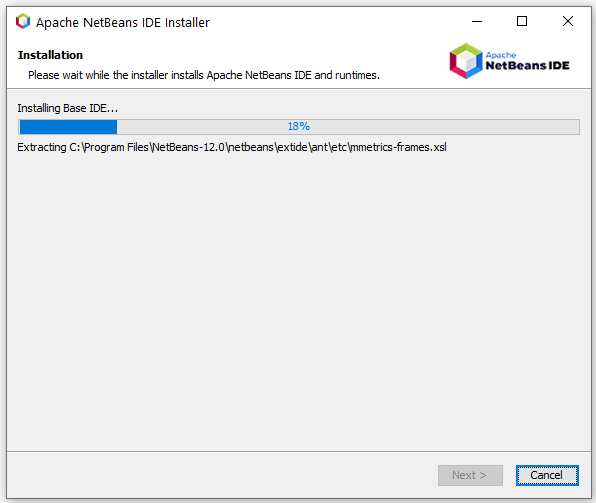
Verify/browse proper JDK location and click next. We have already learnt to install JDK-15.0.2.



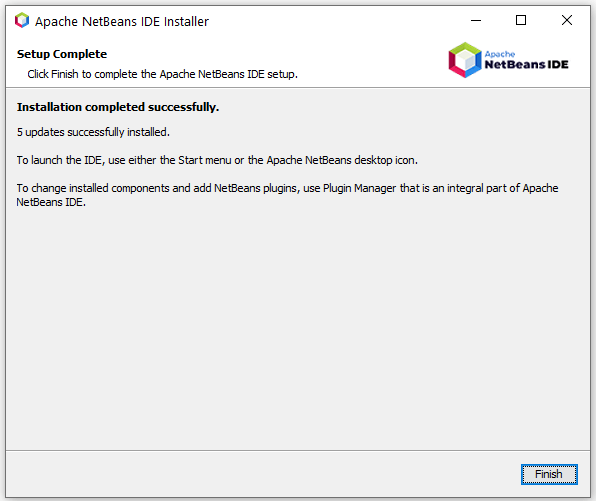
Allow “Check for updates” and click install.



Installation in process.



Installation complete, click finish.



* **Learn to write, compile, execute simple Java programs using notepad**

Open notepad and write following code.

|  |
| --- |
| Program 1.1 |
|  |

In java language, file name should be same as “class” name in which “main” function is called. Further, file extension should be “.java”.

Open command prompt and navigate to location where aziz\_1\_1.java is stored. (D drive)

Write command javac aziz\_1\_1.java to compile this program and to obtain bytecode in the same folder with same name and .class extension.

Write command java aziz\_1\_1 to interpret and execute bytecode to obtain following result.

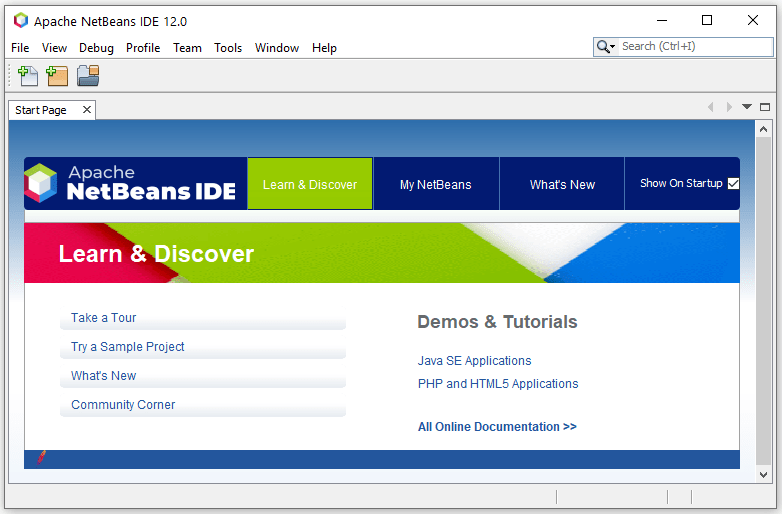
**Output:**

Dr. Abdul Aziz

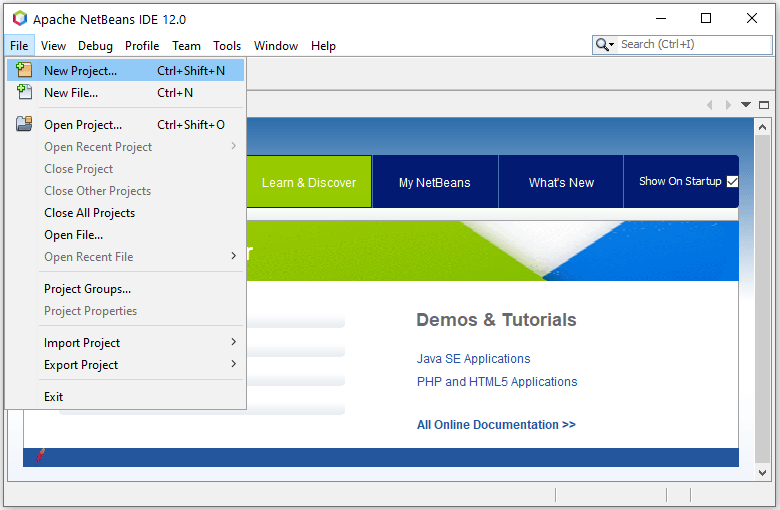
Welcome to Java Programming!

* **Learn to write, run simple Java programs using NetBeans**

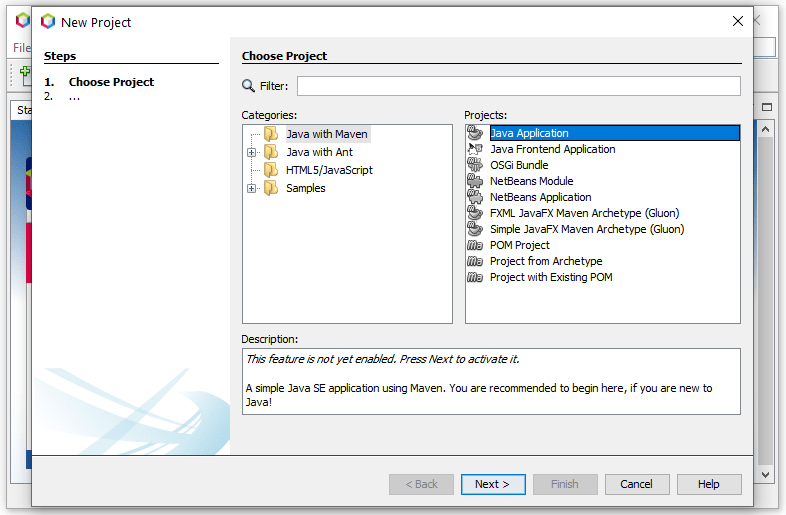
Open NetBeans IDE



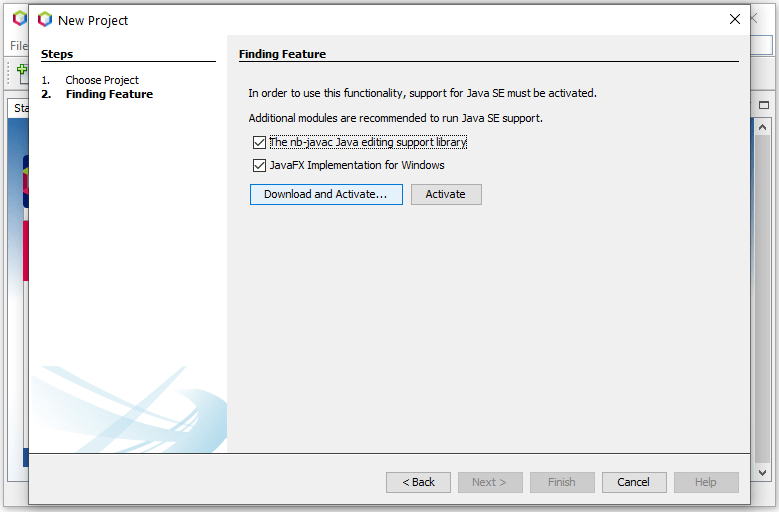
Click on file and then New project



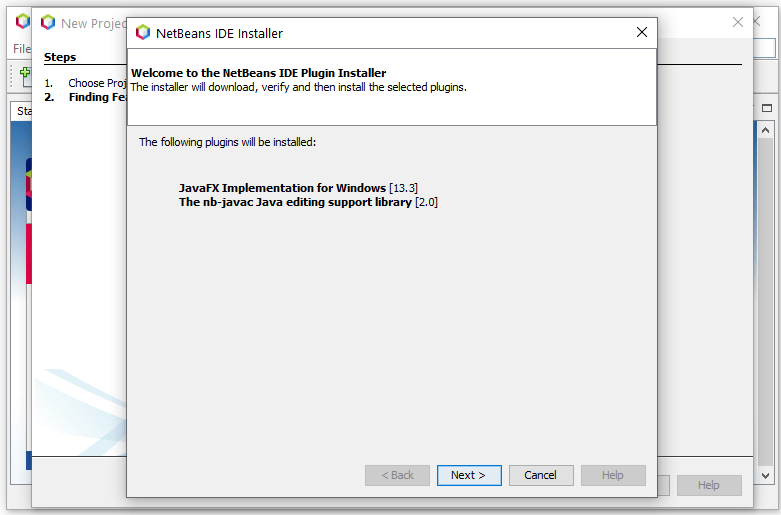
Choose Java with Maven (categories), Java Application (Projects) and click next.



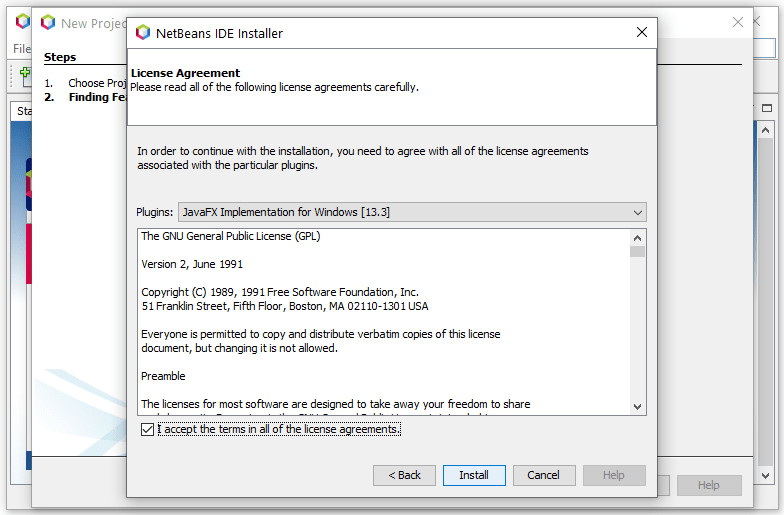
If it is run for the first time, it may ask you activate some feature. Select recommended features to download and activate.



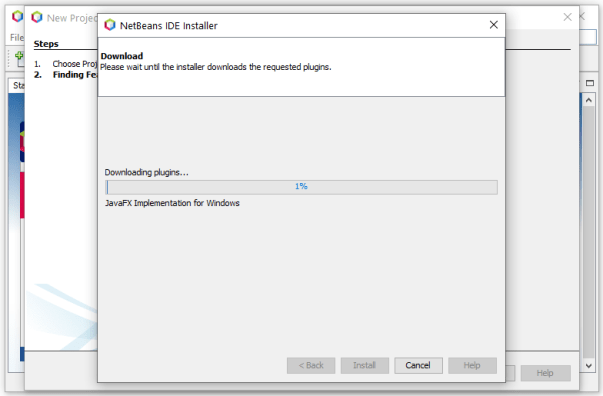
Click next to allow for plugins to be installed.



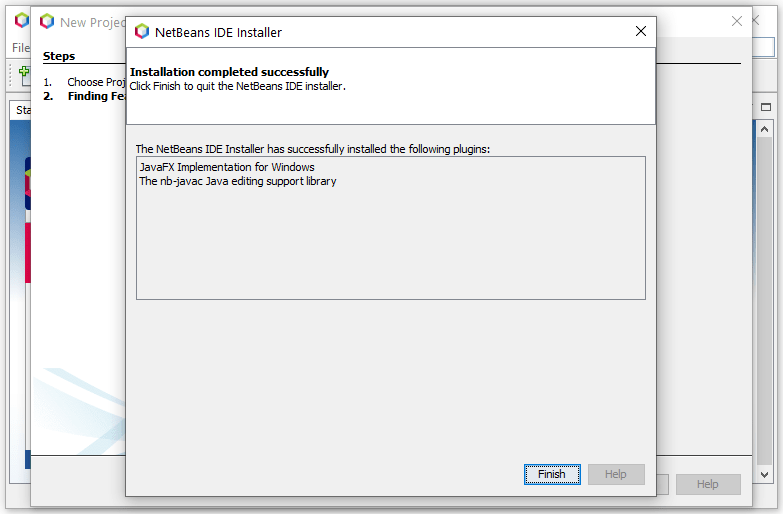
Accept license agreement and click install.



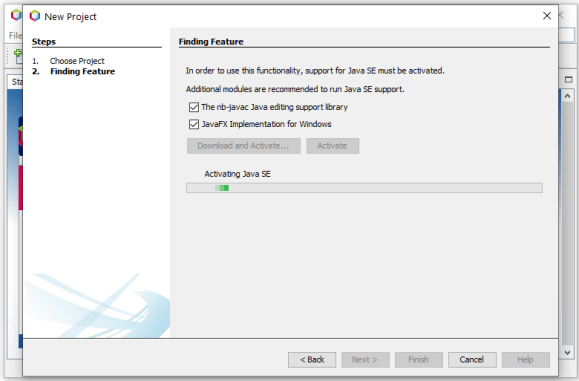
Plugins downloading (wait for few minutes)



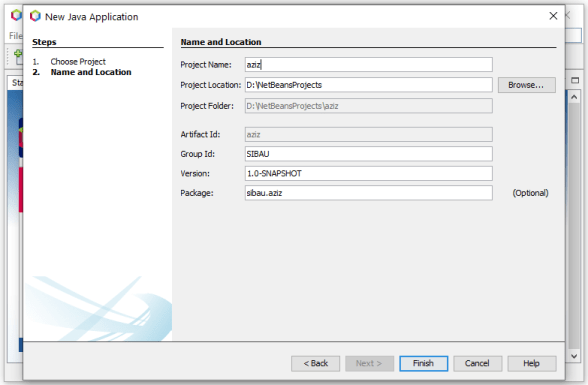
Click finish



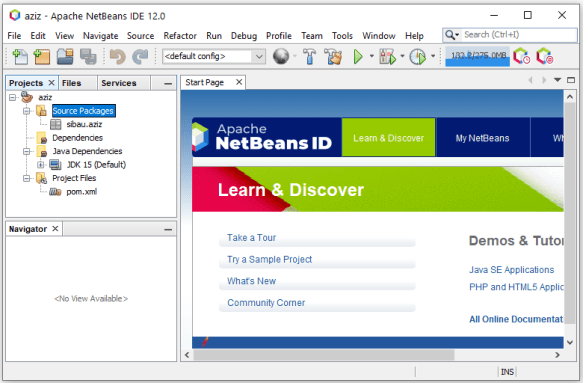
Activation process going on



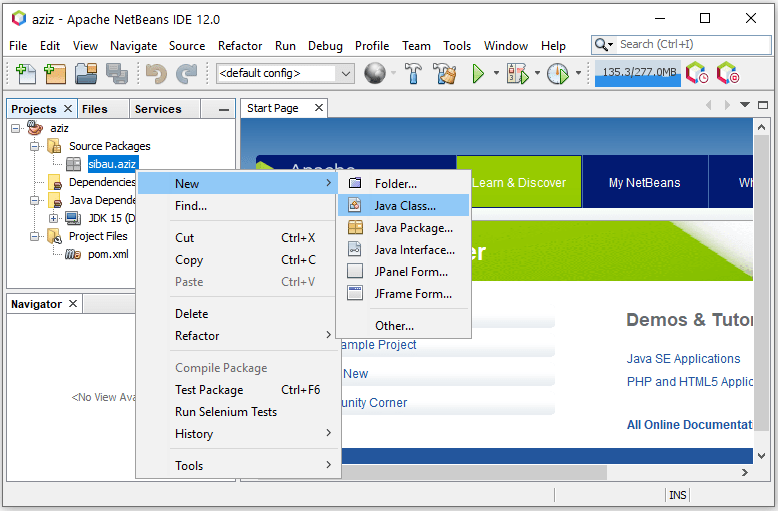
Write project name, browse for project location where you want to save project, write group ID as your company name or website etc and then click finish.



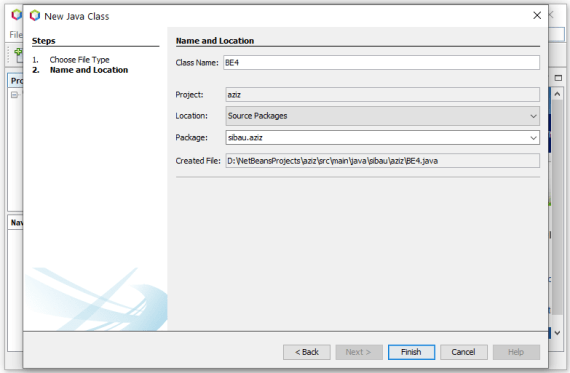
Project aziz is created with various subfolders such as source packages, dependencies, Java dependencies and project files. Right now, we have to just use only source packages folder.



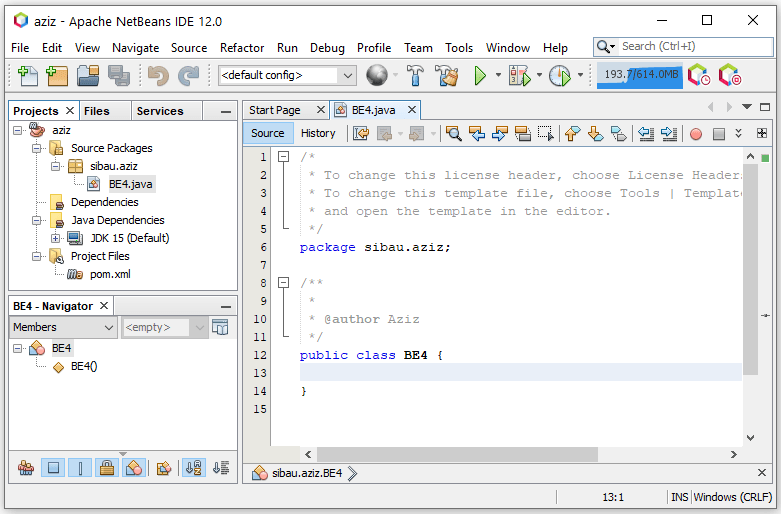
Right click on sibau.aziz, then new, then Java Class.



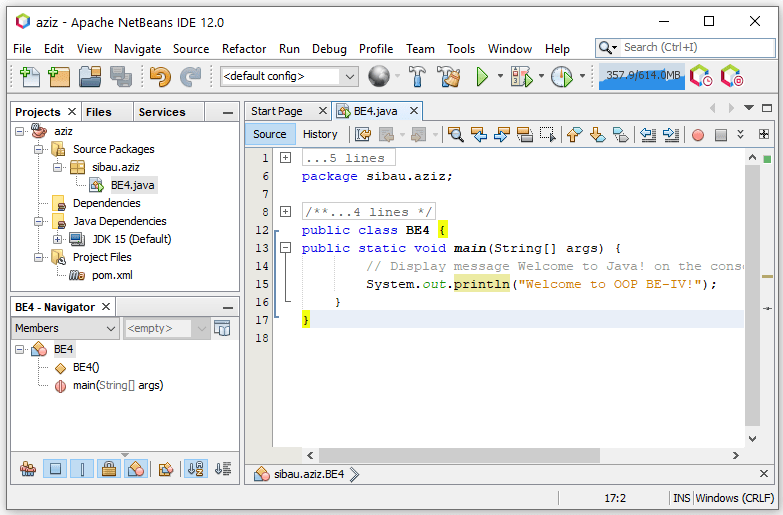
Write class name i.e. BE4 and then click finish. This class will be created at location D:\NetBeansProjects\aziz\src\main\java\sibau\aziz\BE4.java



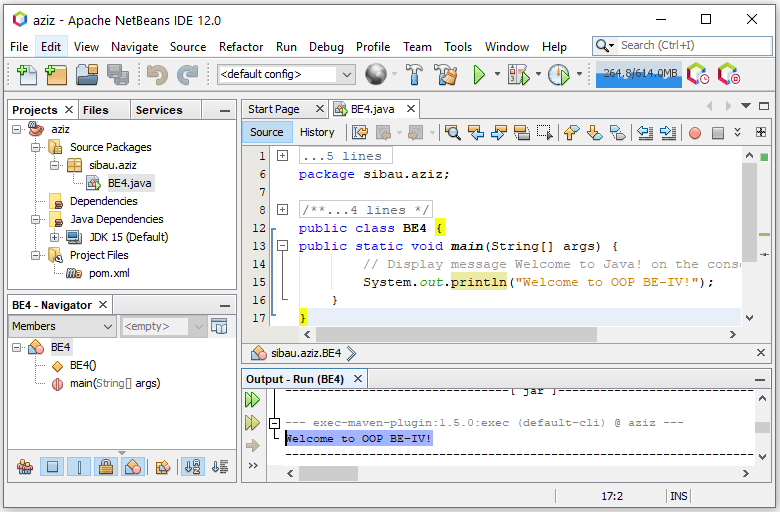
Empty class is created to write code.



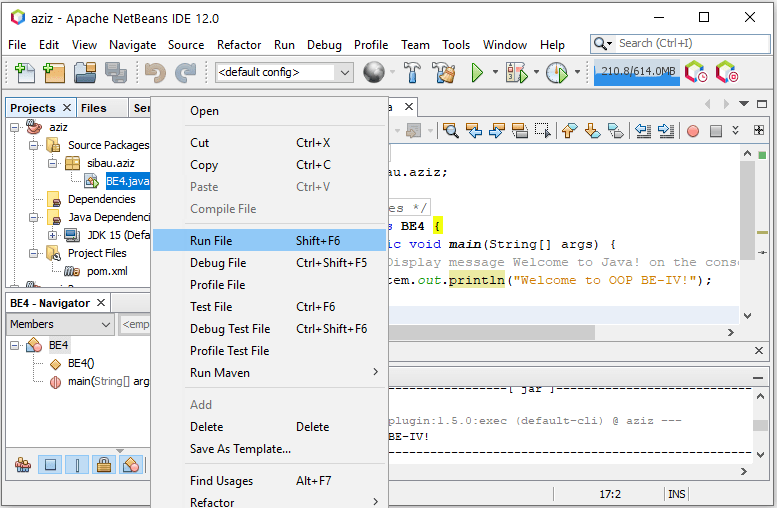
Write simple code to print “Welcome to OOP BE-IV” as an output. Save program and press green triangular play button to view output.



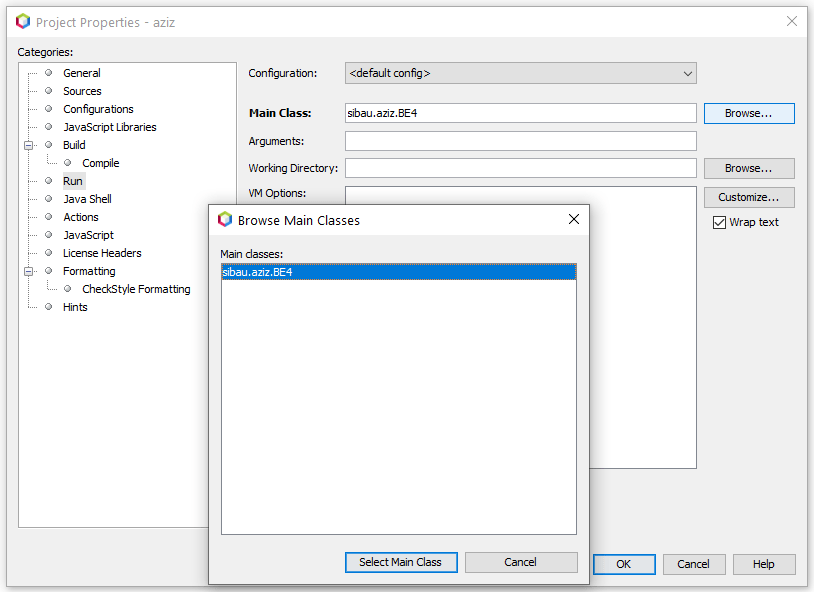
Output can be seen in another window just below the code file.



Note: Once the program is run and if it says that no main classes found then right click on class file BE4.java and then click run.



Or set main class by right clicking project aziz from project window on left side>properties>run>Browse main class



Note: Next time you don’t have to download activate feature. You are ready to create new projects for practice.

Three messages, each on new line.

|  |
| --- |
| Program 1.2 |
| //Code  public class WelcomeWithThreeMessages {  public static void main(String[] args) {  System.out.println("Programming is fun!");  System.out.println("Fundamentals First");  System.out.println("Problem Driven");  }  } |
| Output:  Programming is fun!  Fundamentals First  Problem Driven |

System.out.println🡺used to print some text and moves cursor to new line

System.out.print🡺used to print some text and does not move cursor to new line

Arithmetic operations🡺 println with double quotation prints whatever as it is. Without double quotation it can be used to compute expression. Without double quotation println or print can’t solve characters or strings.

|  |
| --- |
| Program 1.3 |
| //Code  public class ComputeExpression {  public static void main(String[] args) {  System.out.print("(10.5 + 2 \* 3) / (45 - 3.5) = ");  System.out.println((10.5 + 2 \* 3) / (45 - 3.5));  }  } |
| Output:  (10.5 + 2 \* 3) / (45 – 3.5) = 0.39759036144578314 |

Syntax errors: Errors that are detected by the compiler are called *syntax errors* or *compile errors*. Syntax errors result from errors in code construction, such as mistyping a keyword, omitting some necessary punctuation, or using an opening brace without a corresponding closing brace. These errors are usually easy to detect because the compiler tells you where they are and what caused them.

|  |
| --- |
| Program 1.4 |
| public class ShowSyntaxErrors {  public static main(String[] args) {  System.out.println("Welcome to Java);  }  } |
| Output: |

Four errors are reported, but the program actually has two errors:

■ The keyword **void** is missing before **main** in line 2.

■ The string **Welcome to Java** should be closed with a closing quotation mark in line 3.

Since a single error will often display many lines of compile errors, it is a good practice to fix errors from the top line and work downward. Fixing errors that occur earlier in the program may also fix additional errors that occur later.

Runtime errors: *Runtime errors* are errors that cause a program to terminate abnormally. They occur while a program is running if the environment detects an operation that is impossible to carry out. Input mistakes typically cause runtime errors. An *input error* occurs when the program is waiting for the user to enter a value, but the user enters a value that the program cannot handle. For instance, if the program expects to read in a number, but instead the user enters a string, this causes data-type errors to occur in the program.

|  |
| --- |
| Program 1.5 |
| //Code  public class ShowRuntimeErrors {  public static void main(String[] args) {  System.out.println(1 / 0);  }  } |
| Output: |

Logic Errors: *Logic errors* occur when a program does not perform the way it was intended to. Errors of this kind occur for many different reasons. For example, suppose you wrote the program 1.6 to convert Celsius **35** degrees to a Fahrenheit degree. You will get Fahrenheit **67** degrees, which is wrong. It should be **95.0**. In Java, the division for integers is the quotient—the fractional part is truncated—so in Java **9 / 5** is **1**. To get the correct result, you need to use **9.0 / 5**, which results in **1.8**.

In general, syntax and runtime errors are easy to find and easy to correct because those errors can be seen on console. On the other hand, logic errors are difficult to find. We will learn techniques to trace logic errors as well.

|  |
| --- |
| Program 1.6 |
| //Code  public class ShowLogicErrors {  public static void main(String[] args) {  System.out.println("Celsius 35 is Fahrenheit degree ");  System.out.println((9 / 5) \* 35 + 32);  }  } |
| Output:  Celsius 35 is Fahrenheit degree 67 |

Common errors are missing braces, semicolons, quotation marks and misspelling java keywords etc.

Declaring variables, assigning value to variables, defining formula, use of println with +

|  |
| --- |
| Program 1.7 |
| //Code ( + 🡺 string concatenation)  public class ComputeArea {  public static void main(String[] args) {  double radius; // Declare radius  double area; // Declare area  // Assign a radius  radius = 20; // New value is radius  // Compute area  area = radius \* radius \* 3.14159;  // Display results  System.out.println("The area for the circle of radius " +  radius + " is " + area);  }  } |
| Output: |

Using console input with the help of Scanner class to get one value

|  |
| --- |
| Program 1.8 |
| //Code  import java.util.Scanner; // Scanner is in the java.util package  public class ComputeAreaWithConsoleInput {  public static void main(String[] args) {  // Create a Scanner object  Scanner input = new Scanner(System.in);    // Prompt the user to enter a radius  System.out.print("Enter a number for radius: ");  double radius = input.nextDouble();  // Compute area  double area = radius \* radius \* 3.14159;  // Display result  System.out.println("The area for the circle of radius " +  radius + " is " + area);  }  } |
| Output: |

Using console input with the help of Scanner class to get three values

|  |
| --- |
| Program 1.9 |
| //Code  import java.util.Scanner; // Scanner is in the java.util package  public class ComputeAverage {  public static void main(String[] args) {  // Create a Scanner object  Scanner input = new Scanner(System.in);    // Prompt the user to enter three numbers  System.out.print("Enter three numbers: ");  double number1 = input.nextDouble();  double number2 = input.nextDouble();  double number3 = input.nextDouble();  // Compute average  double average = (number1 + number2 + number3) / 3;  // Display result  System.out.println("The average of " + number1 + " " + number2  + " " + number3 + " is " + average);  }  } |
| Output: |

Constant

|  |
| --- |
| Program 1.10 |
| //Code  import java.util.Scanner; // Scanner is in the java.util package  public class ComputeAreaWithConstant {  public static void main(String[] args) {  final double PI = 3.14159; // Declare a constant    // Create a Scanner object  Scanner input = new Scanner(System.in);    // Prompt the user to enter a radius  System.out.print("Enter a number for radius: ");  double radius = input.nextDouble();  // Compute area  double area = radius \* radius \* PI;  // Display result  System.out.println("The area for the circle of radius " +  radius + " is " + area);  }  } |
| Output: |

|  |
| --- |
| Program 1.11 |
| //Convert seconds to minutes & seconds  import java.util.Scanner;  public class DisplayTime {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  // Prompt the user for input  System.out.print("Enter an integer for seconds: ");  int seconds = input.nextInt();    int minutes = seconds / 60; // Find minutes in seconds  int remainingSeconds = seconds % 60; // Seconds remaining  System.out.println(seconds + " seconds is " + minutes +  " minutes and " + remainingSeconds + " seconds");  }} |
| Output: |

|  |
| --- |
| Program 1.12 |
| //Code  import java.util.Scanner;  public class FahrenheitToCelsius {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  System.out.print("Enter a degree in Fahrenheit: ");  double fahrenheit = input.nextDouble();  // Convert Fahrenheit to Celsius  double celsius = (5.0 / 9) \* (fahrenheit - 32);  System.out.println("Fahrenheit " + fahrenheit + " is " +  celsius + " in Celsius");  }  } |
| Output: |

|  |
| --- |
| Program 1.13 |
| //Code  public class ShowCurrentTime {  public static void main(String[] args) {  // Obtain the total milliseconds since midnight, Jan 1, 1970 (Unix Epoch)  long totalMilliseconds = System.currentTimeMillis();  // Obtain the total seconds since midnight, Jan 1, 1970  long totalSeconds = totalMilliseconds / 1000;  // Compute the current second in the minute in the hour  long currentSecond = totalSeconds % 60;  // Obtain the total minutes  long totalMinutes = totalSeconds / 60;  // Compute the current minute in the hour  long currentMinute = totalMinutes % 60;  // Obtain the total hours  long totalHours = totalMinutes / 60;  // Compute the current hour  long currentHour = totalHours % 24;  // Display results  System.out.println("Current time is " + currentHour + ":"  + currentMinute + ":" + currentSecond + " GMT");  }  } |
| Output depends upon current GMT time, PST is GMT+5 hours:  Current time is 17:31:8 GMT |

|  |
| --- |
| Program 1.14 |
| //Code (type casting🡺 from double to int)  import java.util.Scanner;  public class SalesTax {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  System.out.print("Enter purchase amount: ");  double purchaseAmount = input.nextDouble();    double tax = purchaseAmount \* 0.06;  System.out.println("Sales tax is " + (int)(tax \* 100) / 100.0);  }  } |
| Output: Tax amount is truncated to 2 decimal places with the help of (int). |

|  |
| --- |
| Program 1.15 |
| //Code  import java.util.Scanner;  public class ComputeChange {  public static void main(String[] args) {  // Create a Scanner  Scanner input = new Scanner(System.in);  // Receive the amount  System.out.print(  "Enter an amount in double, for example 11.56: ");  double amount = input.nextDouble();  int remainingAmount = (int)(amount \* 100);  // Find the number of one dollars  int numberOfOneDollars = remainingAmount / 100;  remainingAmount = remainingAmount % 100;  // Find the number of quarters in the remaining amount  int numberOfQuarters = remainingAmount / 25;  remainingAmount = remainingAmount % 25;  // Find the number of dimes in the remaining amount  int numberOfDimes = remainingAmount / 10;  remainingAmount = remainingAmount % 10;  // Find the number of nickels in the remaining amount  int numberOfNickels = remainingAmount / 5;  remainingAmount = remainingAmount % 5;  // Find the number of pennies in the remaining amount  int numberOfPennies = remainingAmount;  // Display results  System.out.println("Your amount " + amount + " consists of");  System.out.println(" " + numberOfOneDollars + " dollars");  System.out.println(" " + numberOfQuarters + " quarters ");  System.out.println(" " + numberOfDimes + " dimes");  System.out.println(" " + numberOfNickels + " nickels");  System.out.println(" " + numberOfPennies + " pennies");  }  } |
| Output: |

* javac means java compiler.
* filename.class is java bytecode file.
* If program compiles correctly then there will be no output or error messages on command line, yet there can be runtime errors later.
* [//]🡺 single line comment
* [/\* comments here \*/]🡺 multiline comment
* System.out.println("a" + 1 + 1)🡺a11, but System.out.println("a" + (1 + 1))🡺 a2, and System.out.println(1 + 1)🡺 2 as well.
* public static void main(String[] args) is same as public static void main(String args[])
* Every line that does not begin a new block of code {new block of code}, terminates with semicolon (;).

**Lab exercise**

* Describe in detail with screenshots that how NetBeans debugger works for Java programs
* Write and run all programs from (program 1.1 to program 1.15) either using textpad + command prompt or NetBeans IDE and then paste codes and screenshots of results at the end of this handout in the lab submission section.
* What will be the result of [System.out.println("Welcome\nto\nJava\nProgramming!");]
* What will be the result of [System.out.println("\"in quotes\"");]

1. Modify program [1.15] ComputeChange.java to compute number of lacs, thousands, hundreds, tens, and paisas.
2. (2.6. *Sum the digits in an integer*) Write a program that reads an integer between **0** and **1000** and adds all the digits in the integer. For example, if an integer is **932**, the sum of all its digits is **14**. *Hint*: Use the **%** operator to extract digits, and use the **/** operator to remove the extracted digit. For instance, **932 % 10 = 2** and **932 / 10 = 93**.
3. (2.8. *Current time*) ShowCurrentTime.java, gives a program that displays the current time in GMT. Revise the program so it prompts the user to enter the time zone offset to GMT and displays the time in the specified time zone. Here is a sample run:
4. (2.14. *Health application: computing BMI*) Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing, by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI. Note one pound is **0.45359237** kilograms and one inch is **0.0254** meters.
5. (2.15. *Geometry: distance of two points*) Write a program that prompts the user to enter two points **(x1, y1)** and **(x2, y2)** and displays their distance. Note you can use **Math.pow(a, 0.5)** to compute square root.
6. (2.18. *Print a table*) Write a program that displays table of integer “n” in the standard table format as given below. “n” is equal to last two digits of your CMS ID.

1 x n = n

2 x n = 2n

.

.

10 x n = 2n

1. (2.23. *Cost of driving*) Write a program that prompts the user to enter the distance (in miles) to drive, the fuel efficiency of the car in miles per gallon, and the price per gallon then displays the cost of the trip.

* Verify that:
* From command prompt: command [notepad] and [write] are used to open NotePad and WordPad respectively.
* [notepad filename.java] can be used to directly open program in NotePad using command window.
* If such file does not exist then you will be prompted to create notepad file with name filename.java
* A program written in notepad can be opened in Wordpad using [write filename.java] command in command prompt.
* [type filename.java] opens file contents in command prompt.
* What is the use of [set classpath=class\_path\_here]? i.e. set classpath=c:\code\aziz\
* How can we run Java programs using command prompt and without navigating to actual directory where [filename.java] is stored?

How to change directory in command window:

* Just write cd+complete path and press enter (cd d:\javaprograms\handout1)
* Just write drive letter (d:\) and press enter
* You will in specific directory (d:\javaprograms\handout1)

Online Java IDEs (class name should be changed to “Main” for first two links and MyClass for third link)

<https://www.onlinegdb.com/online_java_compiler>

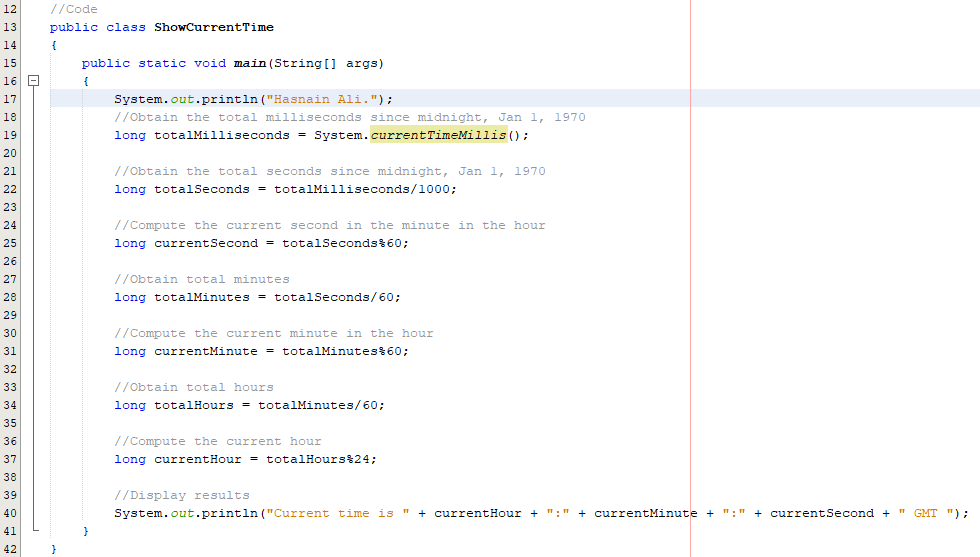
<https://repl.it/languages/java10>

<https://www.jdoodle.com/online-java-compiler/>

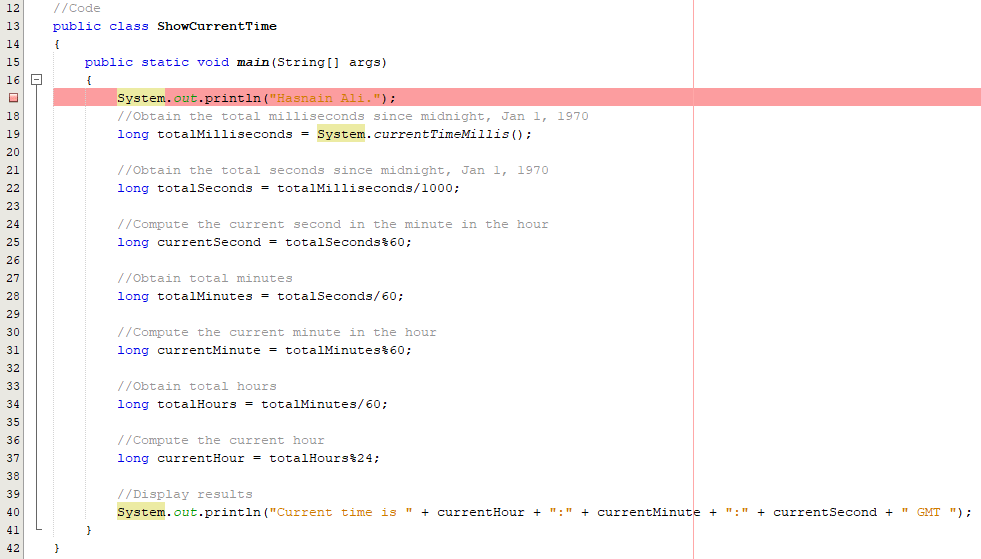
**Lab submission**

* Describe in detail with screenshots that how NetBeans debugger works for Java programs

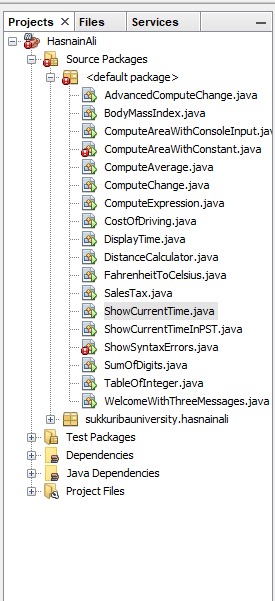
**By using the debugger for the Program 1.13:**



**As the program mainly runs from the main method, a “Breakpoint” will be set on the line number 17 (square shape symbol).**



**From the leftmost window “Projects”, right-click the file “ShowCurrentTime.java” and choose the option “Debug File”.**



**The Debugger will start and there will appear some options like “Debugging” tab on the leftmost side, some options, in a group, of debugger on the top to analyze the program and a status bar to know the output of the program.**

****

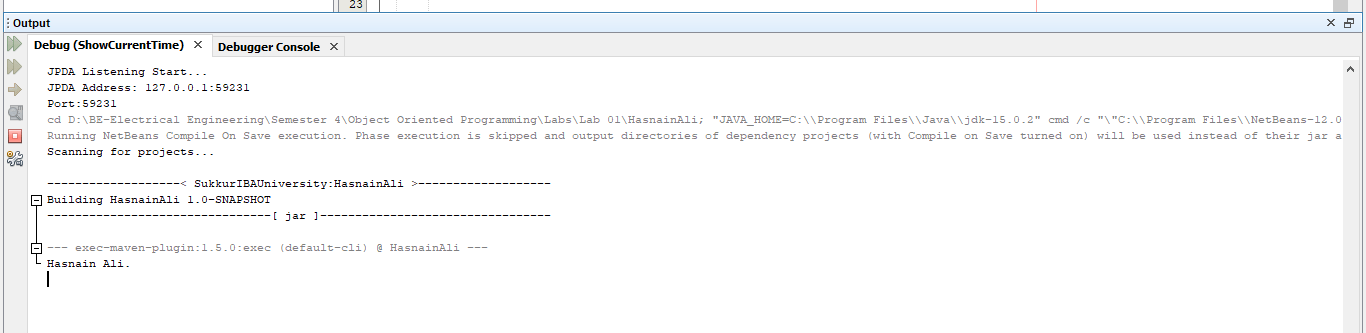
****

****

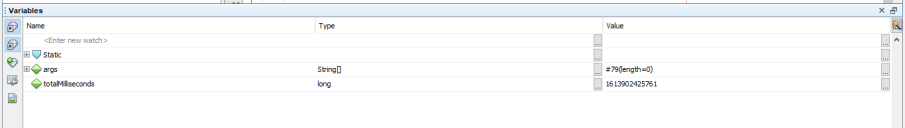
**The “Step Over” option will be pressed one by one to analyze the program from line number 17 to the line number 40 (last line).**



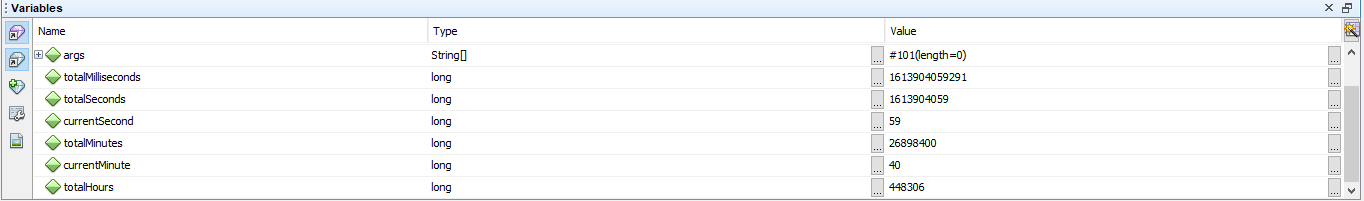
**When “Step Over” button is pressed, the output of first line will be printed which will be seen from “Output” window in the “Status Bar”.**



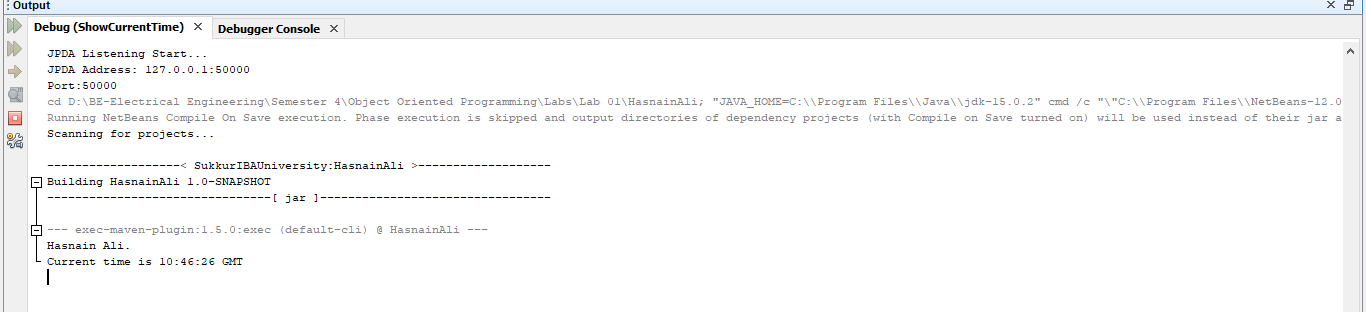
**When the “Step Over” button is pressed again, the debugger will go to the line number 22 where a long variable is declared and initialized. The information of the variable can be seen from the “Variables” window in the “Status Bar”.**



**In this manner, “Step Over” button will be pressed until it reaches line number 37 where the last variable “currentHour” is declared and initialized. Hence, the “Variables” window will collect the information of all the variables of the program line by line.**



**When the “Step Over” button is pressed twice again, it will reach the line number 40 which contains the output statement and will be shown in the “Output” window of the “Status Bar”.**



**In this manner, Debugger analyzes the program line by line along different sections. If there is any variable to be declared to get the input from the user, the input will be taken from the output section, then its information will be shown in the variables section.**

* Write and run all programs from (program 1.1 to program 1.15) either using textpad + command prompt or NetBeans IDE and then paste codes and screenshots of results at the end of this handout in the lab submission section.

Open notepad and write following code.

|  |
| --- |
| Program 1.1 |
| **// Text-printing program.**  **public class hasnain\_1\_1**  **{**  **// main method begins execution of Java application**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali");**  **System.out.println("Welcome to Java Programming!");**  **} // end method main**  **} // end class** |
| Output: |

Three messages, each on new line.

|  |
| --- |
| Program 1.2 |
| **// Code:**  **public class WelcomeWithThreeMessages**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**  **System.out.println("Programming is fun!");**  **System.out.println("Fundamentals First");**  **System.out.println("Problem Driven");**  **}**  **}** |
| Output: |

|  |
| --- |
| Program 1.3 |
| **// Code:**  **public class ComputeExpression**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**  **System.out.println("(10.5 + 2 \* 3)/(45 - 3.5) = ");**  **System.out.println((10.5 + 2 \* 3)/(45 - 3.5));**  **}**  **}** |
| Output: |

|  |
| --- |
| Program 1.4 |
| **public class ShowSyntaxErrors**  **{**  **public static main(String args[])**  **{**  **System.out.println("Hasnain Ali");**  **System.out.println("Welcome to Java);**  **}**  **}** |
| Output: |

|  |
| --- |
| Program 1.5 |
| **//Code**  **public class ShowRuntimeErrors**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**  **System.out.println(1/0);**  **}**  **}** |
| Output: |

|  |
| --- |
| Program 1.6 |
| **//Code**  **public class ShowLogicErrors**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**  **System.out.print("Celsius 35 is Fahrenheit degree ");**  **System.out.print((9/5) \* 35 + 32);**  **}**  **}** |
| Output: |

Declaring variables, assigning value to variables, defining formula, use of println with +

|  |
| --- |
| Program 1.7 |
| **//Code**  **public class ComputeArea**  **{**  **public static void main(String[] args)**  **{**  **System.out.println("Hasnain Ali.");**  **double radius; //Declare radius**  **double area; //Declare area**    **//Assign a radius**  **radius=20; //New value is radius**    **//Compute area**  **area=radius \* radius \* 3.14159;**    **//Display results**  **System.out.println("The area of the circle of radius " + radius + " is " + area);**  **}**  **}** |
| Output: |

Using console input with the help of Scanner class to get one value

|  |
| --- |
| Program 1.8 |
| **//Code**  **import java.util.Scanner; //Scanner is in the java.util package**  **public class ComputeAreaWithConsoleInput**  **{**  **public static void main(String[] args)**  **{**  **System.out.println("Hasnain Ali.");**  **//Create a Scanner object**  **Scanner input = new Scanner(System.in);**    **//Prompt the user to enter a radius**  **System.out.println("Enter a number for radius: ");**  **double radius = input.nextDouble();**    **//Compute area**  **double area = radius \* radius \* 3.14159;**    **//Display result**  **System.out.println("The area for the circle of radius " + radius + " is " + area);**  **}**  **}** |
| Output: |

Using console input with the help of Scanner class to get three values

|  |
| --- |
| Program 1.9 |
| **//Code**  **import java.util.Scanner; //Scanner is in the java.util package**  **public class ComputeAverage**  **{**  **public static void main(String[] args)**  **{**  **System.out.println("Hasnain Ali.");**  **//Create a Scanner object**  **Scanner input = new Scanner(System.in);**    **//Prompt the user to enter three numbers**  **System.out.println("Enter three numbers: ");**  **double number1 = input.nextDouble();**  **double number2 = input.nextDouble();**  **double number3 = input.nextDouble();**    **//Compute Average**  **double average = (number1 + number2 + number3)/3;**    **//Display result**  **System.out.println("The average of " + number1 + ", " + number2 + " and " + number3 + " is " + average);**  **}**  **}** |
| Output: |

Constant

|  |
| --- |
| Program 1.10 |
| **//Code**  **import java.util.Scanner; //Scanner is in the java.util package**  **public class ComputeAreaWithConstant**  **{**  **public static void main(String[] args)**  **{**  **System.out.println("Hasnain Ali.");**    **final double PI = 3.14159; //Declare a constant**    **//Create a Scanner object**  **Scanner input = new Scanner(System.in);**    **//Prompt the user to enter a radius**  **System.out.println("Enter a number for radius: ");**  **double radius = input.nextDouble(4.9);**    **//Compute area**  **double area = radius \* radius \* PI;**    **//Display result**  **System.out.println("The area of the circle of radius " + radius + " is " + area);**  **}**  **}** |
| Output: |

|  |
| --- |
| Program 1.11 |
| **//Code**  **//Convert seconds to minutes and seconds**  **import java.util.Scanner; //Scanner is in the util.java package**  **public class DisplayTime**  **{**  **public static void main(String[] args)**  **{**  **System.out.println("Hasnain Ali.");**  **Scanner input = new Scanner(System.in);**    **//Prompt the user for input**    **System.out.println("Enter an integer for seconds: ");**  **int seconds = input.nextInt();**    **int minutes = seconds/60; //Finding minutes in seconds**  **int remainingSeconds = seconds%60; //Seconds remaining**  **System.out.println(seconds + " seconds is " + minutes + " minutes and "**  **+ remainingSeconds + " seconds");**    **System.out.println();**  **System.out.println();**    **System.out.println("line# \t seconds \t minutes \t remainingSeconds");**  **System.out.println("24 \t " + seconds);**  **System.out.println("28 \t \t \t " + minutes);**  **System.out.println("29 \t \t \t \t \t " + remainingSeconds);**    **}**  **}** |
| Output: |

|  |
| --- |
| Program 1.12 |
| **//Code**  **import java.util.Scanner;**  **public class FahrenheitToCelsius**  **{**  **public static void main(String[] args)**  **{**  **System.out.println("Hasnain Ali.");**  **Scanner input = new Scanner(System.in);**    **System.out.println("Enter a degree in Fahrenheit: ");**  **double fahrenheit = input.nextDouble();**    **//Convert Fahrenheit to Celsius**  **double celsius = (5.0/9) \* (fahrenheit - 32);**  **System.out.println("Fahrenheit " + fahrenheit + " is " + celsius + " in Celsius");**    **System.out.println();**  **System.out.println();**  **System.out.println();**    **System.out.println("line# \t fahrenheit \t celsius");**  **System.out.println("22 \t " + fahrenheit);**  **System.out.println("26 \t \t \t " + celsius);**  **}**  **}** |
| Output: |

|  |
| --- |
| Program 1.13 |
| **//Code**  **public class ShowCurrentTime**  **{**  **public static void main(String[] args)**  **{**  **System.out.println("Hasnain Ali.");**  **//Obtain the total milliseconds since midnight, Jan 1, 1970**  **long totalMilliseconds = System.currentTimeMillis();**    **//Obtain the total seconds since midnight, Jan 1, 1970**  **long totalSeconds = totalMilliseconds/1000;**    **//Compute the current second in the minute in the hour**  **long currentSecond = totalSeconds%60;**    **//Obtain total minutes**  **long totalMinutes = totalSeconds/60;**    **//Compute the current minute in the hour**  **long currentMinute = totalMinutes%60;**    **//Obtain total hours**  **long totalHours = totalMinutes/60;**    **//Compute the current hour**  **long currentHour = totalHours%24;**    **//Display results**  **System.out.println("Current time is " + currentHour + ":" + currentMinute + ":" + currentSecond + " GMT ");**  **}**  **}** |
| Output: |

|  |
| --- |
| Program 1.14 |
| **//Code**  **import java.util.Scanner; //Scanner is in the java.util package**  **public class SalesTax**  **{**  **public static void main(String[] args)**  **{**  **System.out.println("Hasnain Ali.");**    **Scanner input = new Scanner(System.in);**    **System.out.println("Enter purchase amount: ");**  **double purchaseAmount = input.nextDouble();**    **double tax = purchaseAmount \* 0.06;**  **System.out.println("Sales tax is $" + (int)(tax \* 100)/100.0);**    **System.out.println();**  **System.out.println();**  **System.out.println();**    **System.out.println("line# \t purchaseAmount \t tax \t Output");**  **System.out.println("22 \t " + purchaseAmount);**  **System.out.println("24 \t \t \t \t" + tax);**  **System.out.println("25 \t \t \t \t \t " + (int)(tax \* 100)/100.0);**  **}**  **}** |
| Output: Tax amount is truncated to 2 decimal places with the help of (int). |

|  |
| --- |
| Program 1.15 |
| **//Code**  **import java.util.Scanner; //Scanner is in the java.util package**  **public class ComputeChange**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**    **//Create a Scanner**  **Scanner input = new Scanner(System.in);**    **//Receive the amount**  **System.out.println("Enter an amount in double, for example 11.56: ");**  **double amount = input.nextDouble();**    **int remainingAmount = (int)(amount \* 100);**    **//Find the number of one dollar**  **int numberOfOneDollar = remainingAmount/100;**  **remainingAmount = remainingAmount%100;**    **//Find the number of quarters in the remaining amount**  **int numberOfQuarters = remainingAmount/25;**  **remainingAmount = remainingAmount%25;**    **//Find the number of dimes in the remaining amount**  **int numberOfDimes = remainingAmount/10;**  **remainingAmount = remainingAmount%10;**    **//Find the number of nickels in the remaining amount**  **int numberOfNickels = remainingAmount/5;**  **remainingAmount = remainingAmount%5;**    **//Find the number of pennies in the remaining amount**  **int numberOfPennies = remainingAmount;**    **//Display results**  **System.out.println("Your amount " + amount + " consists of ");**  **System.out.println(" " + numberOfOneDollar + " dollars");**  **System.out.println(" " + numberOfQuarters + " quarters");**  **System.out.println(" " + numberOfDimes + " dimes");**  **System.out.println(" " + numberOfNickels + " nickels");**  **System.out.println(" " + numberOfPennies + " pennies");**    **}**  **}** |
| Output: |

* What will be the result of [System.out.println("Welcome\nto\nJava\nProgramming!");]

**“\n” gives next line.**



* What will be the result of [System.out.println("\"in quotes\"");]

**“\” “\” gives the string in quotations.**



1. Modify program [1.15] ComputeChange.java to compute number of lacs, thousands, hundreds, tens, and paisas.

|  |
| --- |
| Exercise 1.1 |
| **//Code**  **import java.util.Scanner; //Scanner is in the util.java package**  **public class AdvancedComputeChange**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**      **//Creating a Scanner object**  **Scanner input = new Scanner(System.in);**    **//Get the amount**  **System.out.println("Enter an amount in double. For example, 627963.18");**  **double amount = input.nextDouble();**    **//Converting the entered amount into a complete integer**  **int remainingAmount = (int)(amount \* 100);**    **//Getting number of lacs**  **int numberOfLacs = remainingAmount/10000000;**  **remainingAmount %= 10000000;**    **//Getting number of thousands**  **int numberOfThousands = remainingAmount/100000;**  **remainingAmount %= 100000;**    **//Getting number of hundreds**  **int numberOfHundreds = remainingAmount/10000;**  **remainingAmount %= 10000;**    **//Getting number of tens**  **int numberOfTens = remainingAmount/1000;**  **remainingAmount %= 1000;**    **//Getting number of ones**  **int numberOfOnes = remainingAmount/100;**  **remainingAmount %= 100;**    **//Getting number of paisas**  **int numberOfPaisas = remainingAmount;**    **//Display calculations**  **System.out.println("The entered amount " + amount + " contains ");**  **System.out.println(" " + numberOfLacs + " lacs");**  **System.out.println(" " + numberOfThousands + " thousands");**  **System.out.println(" " + numberOfHundreds + " hundreds");**  **System.out.println(" " + numberOfTens + " tens");**  **System.out.println(" " + numberOfOnes + " ones");**  **System.out.println(" " + numberOfPaisas + " paisas");**        **}**  **}** |
| Output: |

1. (2.6. *Sum the digits in an integer*) Write a program that reads an integer between **0** and **1000** and adds all the digits in the integer. For example, if an integer is **932**, the sum of all its digits is **14**.

|  |
| --- |
| Exercise 1.2 |
| **//Code**  **import java.util.Scanner;**  **public class SumOfDigits**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**  **Scanner input = new Scanner(System.in);**    **System.out.println("Enter an integer: ");**      **long integer = input.nextLong();**  **integer = Math.abs(integer); //In case of entered negative number**    **long number = integer; //To show the integer in the result**  **number = Math.abs(number);**    **long quotient=0;**  **long remainder=0;**  **long sum=0;**    **while(!(integer>0 && integer<=9))**  **{**  **quotient = integer%10;**  **remainder = integer/10;**  **sum += quotient;**  **integer = remainder;**  **}**  **sum += integer; //Here, the while loop gives away the remaining one's digit of the entered integer**  **//which is added in the sum outside the loop's block.**    **System.out.println("The sum of the digits of " + number + " is " + sum);**  **}**  **}** |
| Output: |

1. (2.8. *Current time*) ShowCurrentTime.java, gives a program that displays the current time in GMT. Revise the program so it prompts the user to enter the time zone offset to GMT and displays the time in the specified time zone. Here is a sample run:

|  |
| --- |
| Exercise 1.3 |
| **//Code**  **import java.util.Scanner;**  **public class ShowCurrentTimeInPST**  **{**  **public static void main(String[] args)**  **{**  **System.out.println("Hasnain Ali.");**    **//Declaring a Scanner object**  **Scanner input = new Scanner(System.in);**      **//Obtain the total milliseconds since midnight, Jan 1, 1970**  **long totalMilliseconds = System.currentTimeMillis();**    **//Obtain the total seconds since midnight, Jan 1, 1970**  **long totalSeconds = totalMilliseconds/1000;**    **//Compute the current second in the minute in the hour**  **long currentSecond = totalSeconds%60;**    **//Obtain total minutes4**  **long totalMinutes = totalSeconds/60;**    **//Compute the current minute in the hour**  **long currentMinute = totalMinutes%60;**    **//Obtain total hours**  **long totalHours = totalMinutes/60;**    **//Compute the current hour**  **long currentHour = totalHours%24;**  **System.out.println("Current time is " + currentHour + ":" + currentMinute + ":" + currentSecond + " GMT \n");**    **System.out.println("Enter an offset of 4 to convert it into PST: ");**  **int offset = input.nextInt();**    **if(offset!=4)**  **{**  **System.out.println("Please run the program again and give correct input!");**  **}**  **else**  **{**  **currentHour += offset;**  **if(currentHour == 24)**  **{**  **currentHour -= 12; //For appropriate 24 hours format**  **}**  **else if(currentHour>24)**  **{**  **currentHour -= 24; //For appropriate 24 hours format**  **}**    **//Display results**  **System.out.println("\nCurrent time is " + currentHour + ":" + currentMinute + ":" + currentSecond + " PST ");**  **}**    **}**  **}** |
| Output: |

1. (2.14. *Health application: computing BMI*) Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing, by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI. Note one pound is **0.45359237** kilograms and one inch is **0.0254** meters.

|  |
| --- |
| Exercise 1.4 |
| **//Code**  **import java.util.Scanner;**  **public class BodyMassIndex**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**  **Scanner input = new Scanner(System.in);**    **System.out.println("Enter weight in pounds: ");**  **double weight = input.nextDouble();**    **System.out.println("Enter height in inches:" );**  **double height = input.nextDouble();**    **weight \*= 0.45359237; //Converting into kilograms**  **height \*= 0.0254; //Converting into meters**    **double bmi = weight/(Math.pow(height,2));**    **System.out.println("Body Mass Index (BMI) is " + bmi + " kg/sq.meter");**  **}**  **}** |
| Output: |

1. (2.15. *Geometry: distance of two points*) Write a program that prompts the user to enter two points **(x1, y1)** and **(x2, y2)** and displays their distance. Note you can use **Math.pow(a, 0.5)** to compute square root.

|  |
| --- |
| Exercise 1.5 |
| **//Code**  **import java.util.Scanner;**  **public class DistanceCalculator**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**  **Scanner input = new Scanner(System.in);**    **System.out.println("Enter the abscissa of first point: ");**  **double x1 = input.nextDouble();**    **System.out.println("Enter the ordinate of first point: ");**  **double y1 = input.nextDouble();**    **System.out.println();**    **System.out.println("Enter the abscissa of second point: ");**  **double x2 = input.nextDouble();**    **System.out.println("Enter the ordinate of second point: ");**  **double y2 = input.nextDouble();**    **double firstTerm = Math.pow(x2 - x1,2);**  **double secondTerm = Math.pow(y2 - y1, 2);**    **double distance = Math.pow(firstTerm + secondTerm,0.5);**    **System.out.println("The distance between (" + x1 + "," + y1 + ")" + " and " + "(" + x2 + "," + y2 + ")");**  **System.out.println(" is " + distance + " units");**  **}**    **}** |
| Output: |

1. (2.18. *Print a table*) Write a program that displays table of integer “n” in the standard table format as given below. “n” is equal to last two digits of your CMS ID.

1 x n = n

2 x n = 2n

.

.

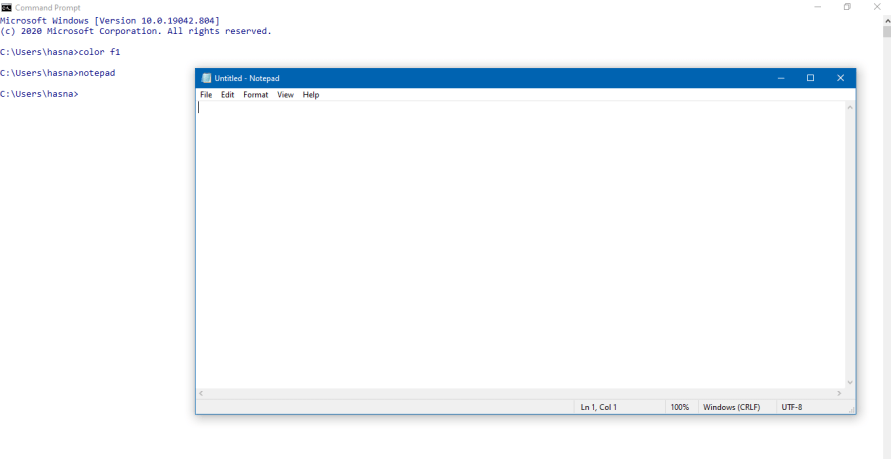
10 x n = 2n

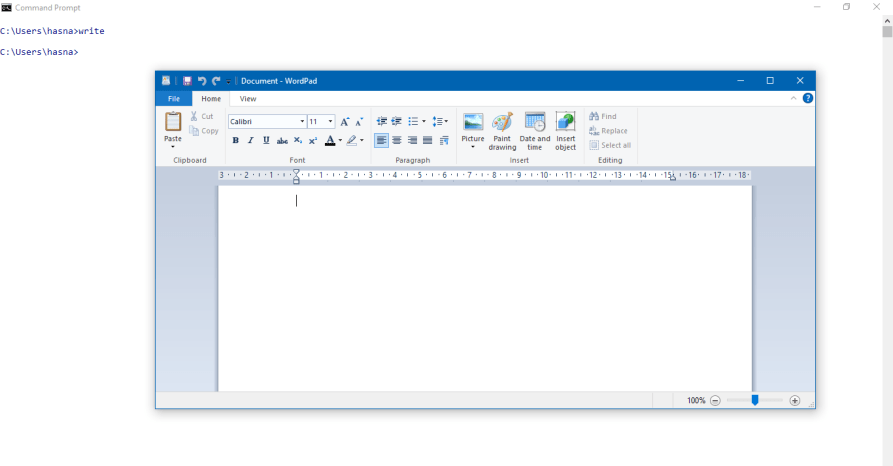
|  |
| --- |
| Exercise 1.6 |
| **//Code**  **public class TableOfInteger**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**  **System.out.println();**    **long n = 28;**    **for(int i=1;i<=10;i++)**  **{**  **System.out.println(i + " x " + n + " = " + (i\*n));**  **}**  **}**  **}** |
| Output: |

1. (2.23. *Cost of driving*) Write a program that prompts the user to enter the distance (in miles) to drive, the fuel efficiency of the car in miles per gallon, and the price per gallon then displays the cost of the trip.

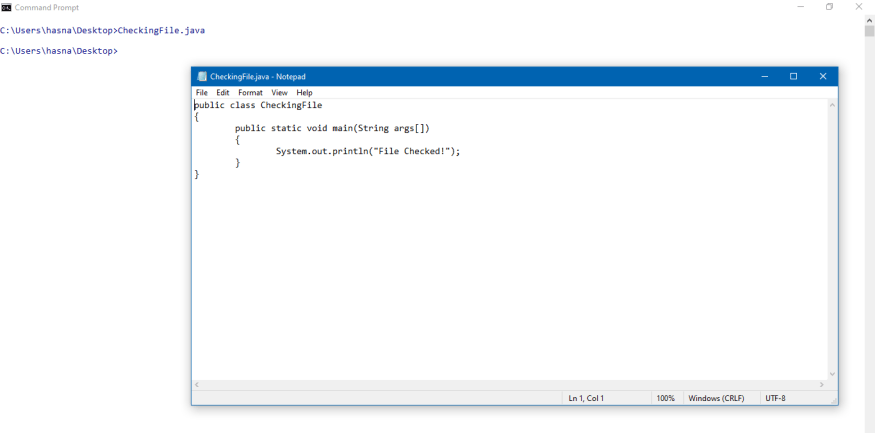
|  |
| --- |
| Exercise 1.7 |
| **//Code**  **import java.util.Scanner;**  **public class CostOfDriving**  **{**  **public static void main(String args[])**  **{**  **System.out.println("Hasnain Ali.");**    **Scanner input = new Scanner(System.in);**    **System.out.println("Enter the distance to be covered by the car (in miles): ");**  **double distance = input.nextDouble();**    **System.out.println("Enter the fuel efficiency of the car in miles per gallon (mpg): ");**  **double fuelEfficiencyOfCar = input.nextDouble();**    **System.out.println("Enter the price of one gallon in dollars ($): ");**  **double priceOfOneGallon = input.nextDouble();**    **double noOfGallons = distance / fuelEfficiencyOfCar;**  **double costOfTheTrip = noOfGallons \* priceOfOneGallon;**    **System.out.println("The cost of the trip will be $" + costOfTheTrip);**    **}**  **}** |
| Output: |

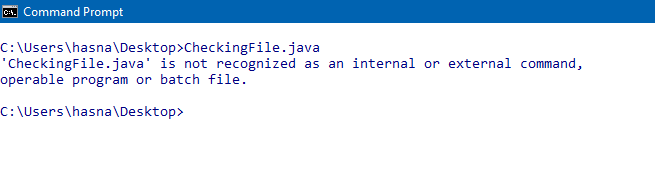
* From command prompt: command [notepad] and [write] are used to open NotePad and WordPad respectively.



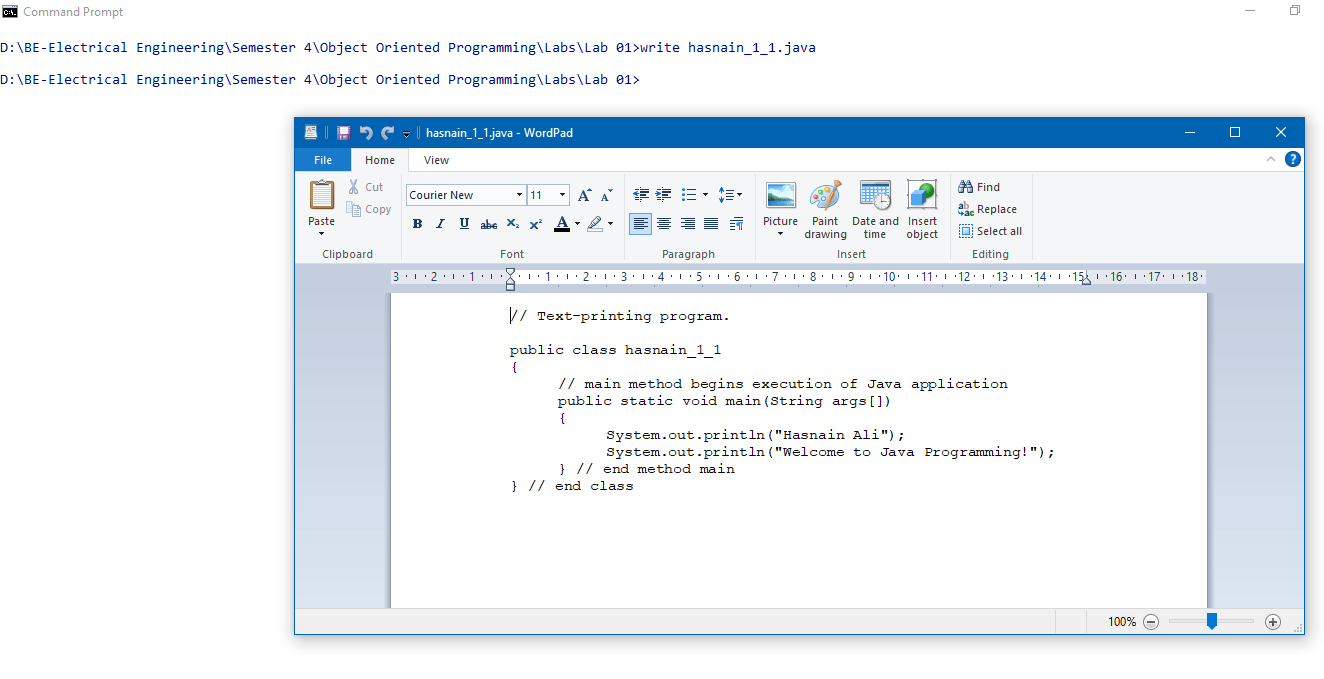


* [notepad filename.java] can be used to directly open program in NotePad using command window.
* If such file does not exist then you will be prompted to create notepad file with name filename.java

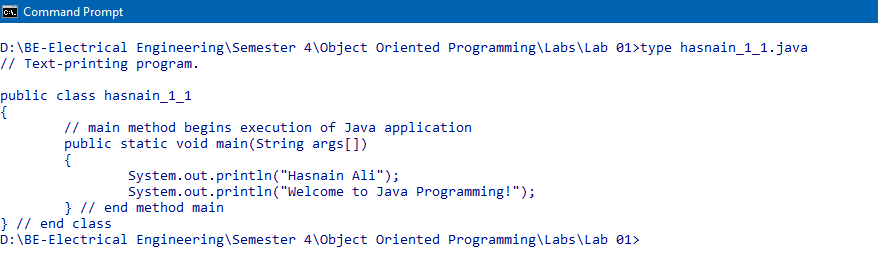




* A program written in notepad can be opened in Wordpad using [write filename.java] command in command prompt.



* [type filename.java] opens file contents in command prompt.



* What is the use of [set classpath=class\_path\_here]? i.e. set classpath=c:\code\aziz\

**“Classpath” is used to locate the path where .class file is stored.**

* How can we run Java programs using command prompt and without navigating to actual directory where [filename.java] is stored?

**Using “JShell” command on command prompt.**