

Introduction to NetBeans

To complete the work sheets, you will use the Apache NetBeans (12.5) as IDE (Integrated Development Environment) and JDK (Java Development Kit) Version 16 as the development Environment.

Virtual machines which you receive already installed above software's

1. Open NetBeans IDE

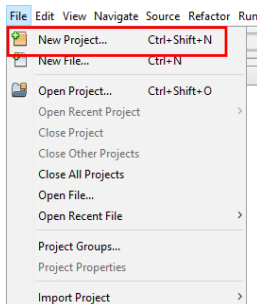
Start Menu -> NetBeans



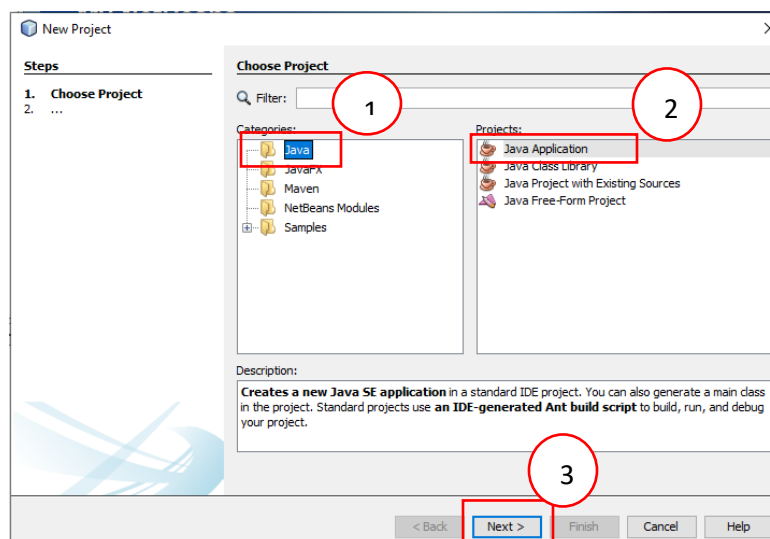
(IF there is a desktop shortcut you can use that)

2. Create New Java Project

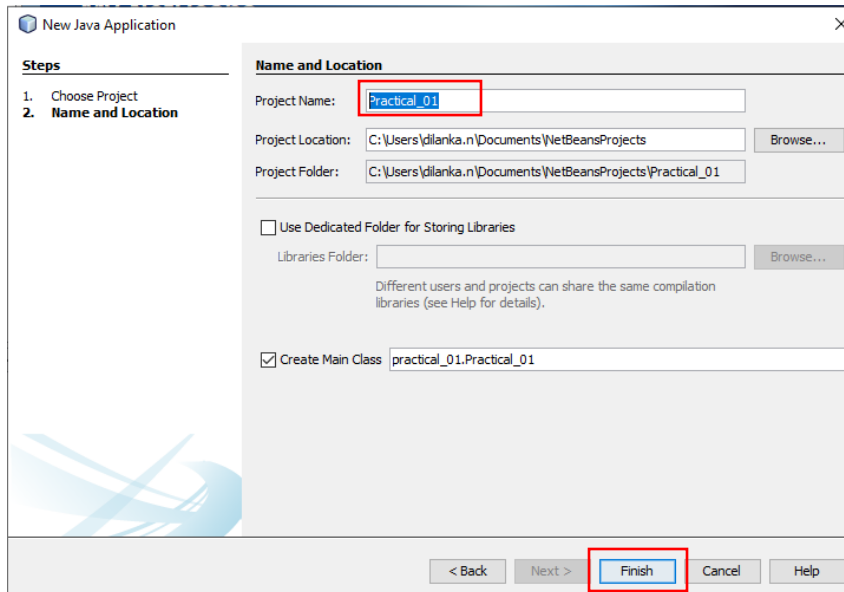
File -> New Project



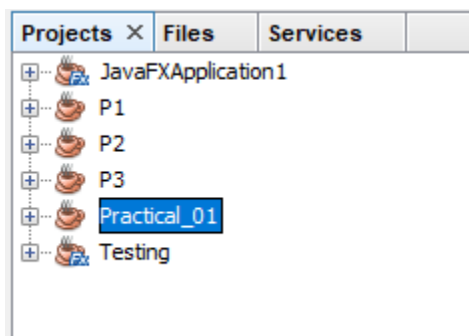
After you will have a interface to select project type as below.



- Select **Java** in Categories Column,
- Select **Java Application** in Project Column,
- Click **Next**



- Then Give an appropriate name for you project, **EX: Practical_01**
- Click **Finish**.

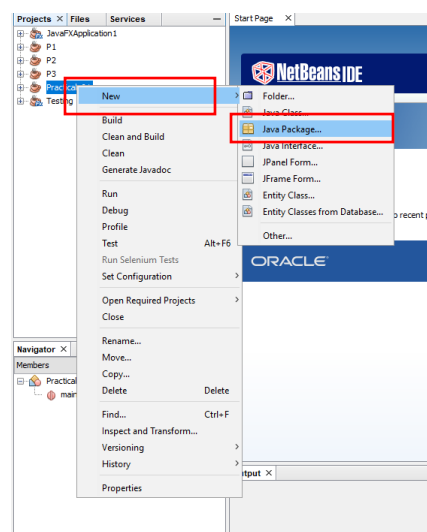


- Now you can see the project which you created under the **Project Explorer**

3. Create Java Packages and Classes

To create packages right click on your project in project Explorer, Then Select,

New → New Packages



MDTU JAVA Workshop Worksheet 01

New Java Package

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

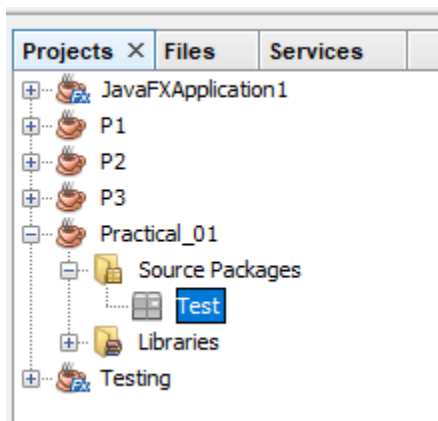
Package Name:

Project:

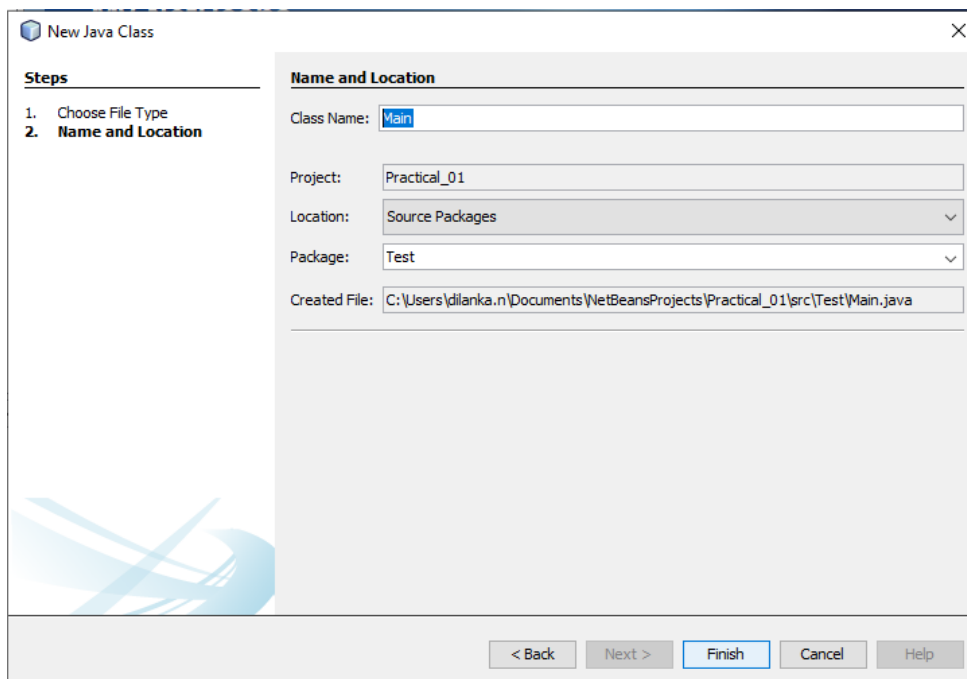
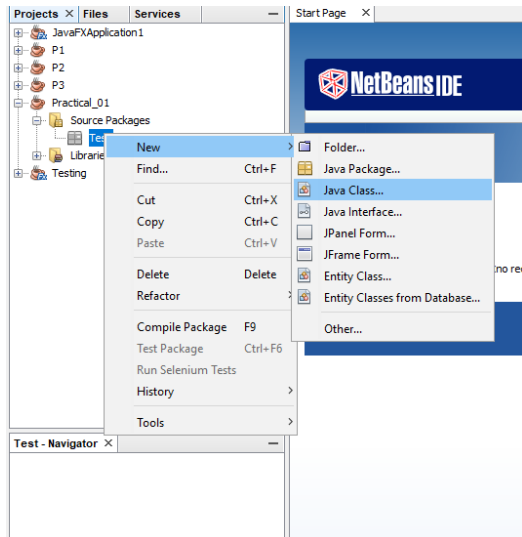
Location:

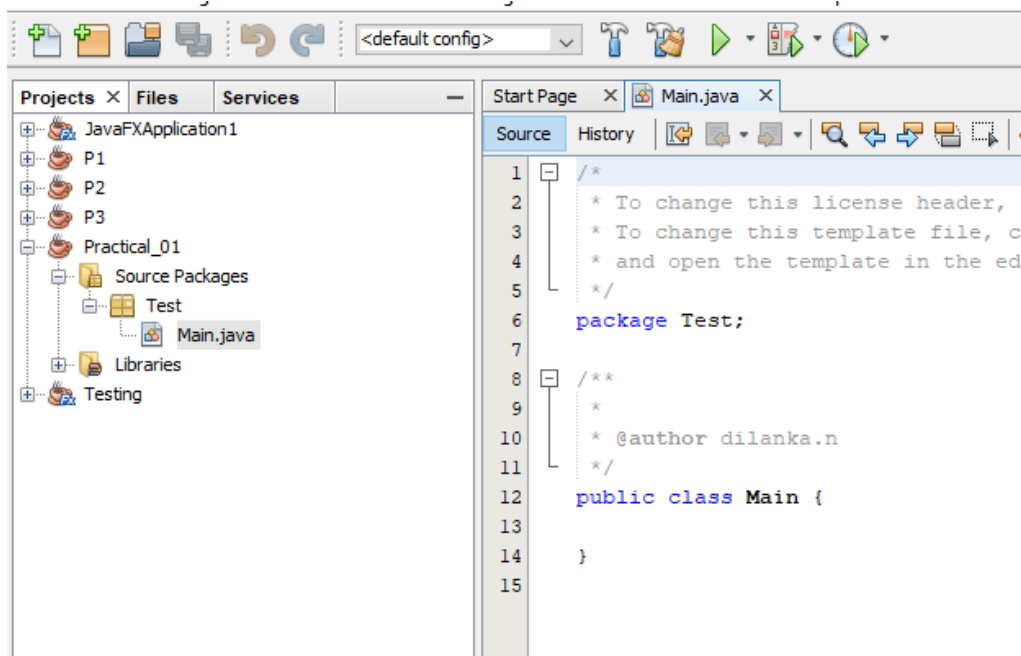
Created Folder: C:\Users\dilanka.n\Documents\NetBeansProjects\Practical_01\src\Test

< Back Next > **Finish** Cancel Help



To create packages right click on your project in project Explorer, Then Select,
Right Click on the package, **New -> Java class**





Worksheet 01

The objective of this practical is to build a console-based file handling application using Java, which could to basic file handling such as create, write, read, and list files. You are given set of activities and the sample outputs to your reference.

Activity 1

Write a program to display the main menu in the console where user can input the action as a number through the keyboard.

Sample output: -

```
Simple File Handling Application
Press (1) to create a file:
Press (2) to write to a file:
Press (3) to read from a file:
Press (4) to list files in folder:
Enter Option: *
```

Activity 2

Modify the above program where if user enter 1 as the option, to create a file by taking the file name as a keyboard input. If the file name already exists, “Error: File Already Exists” should be displayed and program terminates. If the file does not exist, a file should be created in the same location where you create the project with the user given name. After creating the file, “File created successfully” message should be display to the user.

Sample output: -

```
Simple File Handling Application
Press (1) to create a file:
Press (2) to write to a file:
Press (3) to read from a file:
Press (4) to list files in folder:
Enter Option: 1
Enter File Name: Test.txt
Successfully Created
```

Activity 3

Modify the same program where if user enter 2 as the option, to write to the file.

User should be able to give the file name as a keyboard input when writing to the file. If the file name is not existing “Error: File Does Not Exist” should be displayed and program terminates.

If the given file name is already existed, user should be able to type the words where it needs to write, until user type exit. When user type exit, the content should be written into the file and program should terminates by displaying the message as “File Closed”.

Sample output: -

```
Simple File Handling Application
Press (1) to create a file:
Press (2) to write to a file:
Press (3) to read from a file:
Press (4) to list files in folder:
Enter Option: 2
Enter File Name: test.txt
Enter text (enter exit to quit): hello world
Enter text (enter exit to quit): i love JAVA
Enter text (enter exit to quit): exit
File Closed
```

Activity 4

Modify the same program where if user enter 3 as the option, to read from the file.

User should be able to give the file name as a keyboard input when reading from the file. If the file name is not existing “Error: File Does Not Exist” should be displayed and program terminates.

If the given file name is already existed, the content of the file should be displayed in the console output and program should terminates by displaying the message as “File Closed”.

Sample output: -

```
Simple File Handling Application
Press (1) to create a file:
Press (2) to write to a file:
Press (3) to read from a file:
Press (4) to list files in folder:
Enter Option: 3
Enter File Name: test.txt
hello world
i love JAVA
```

```
File Closed
```

Activity 5

Modify the same program where if user enter 4 as the option, to list down all the files in the folder.

User should be able to give the folder path as the keyboard input. Then the program should be able to list all the files in the folder in the console output.

Sample output: -

```
Simple File Handling Application
Press (1) to create a file:
Press (2) to write to a file:
Press (3) to read from a file:
Press (4) to list files in folder:
Enter Option: 4
Enter the path: C:\Users\HP\Documents\NetBeansProjects\P1_Updated
File build.xml
File good.txt
File manifest.mf
File mytest
File test.txt
```