Object Oriented Programming Lab

CSE 1206

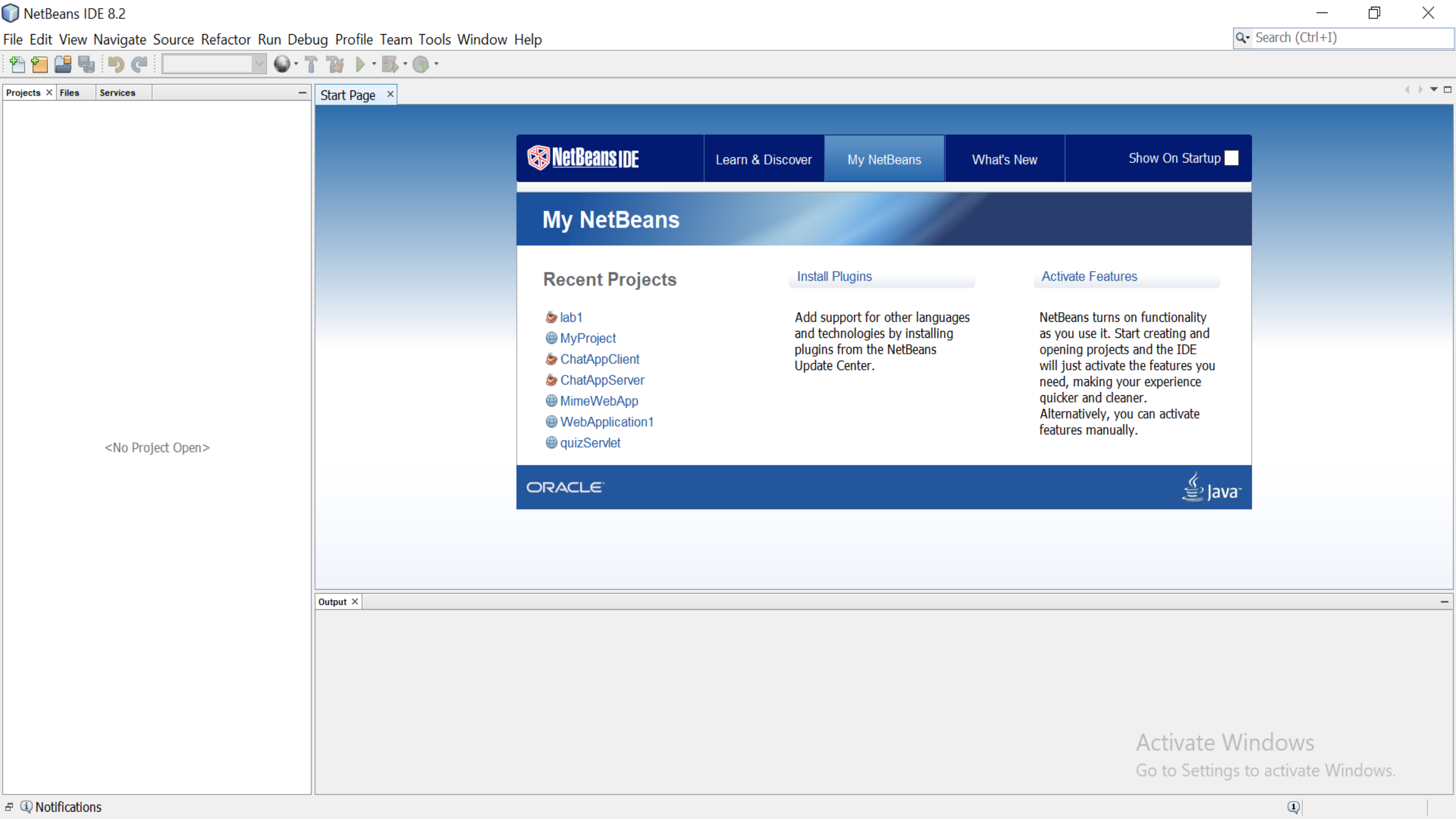
LAB 2

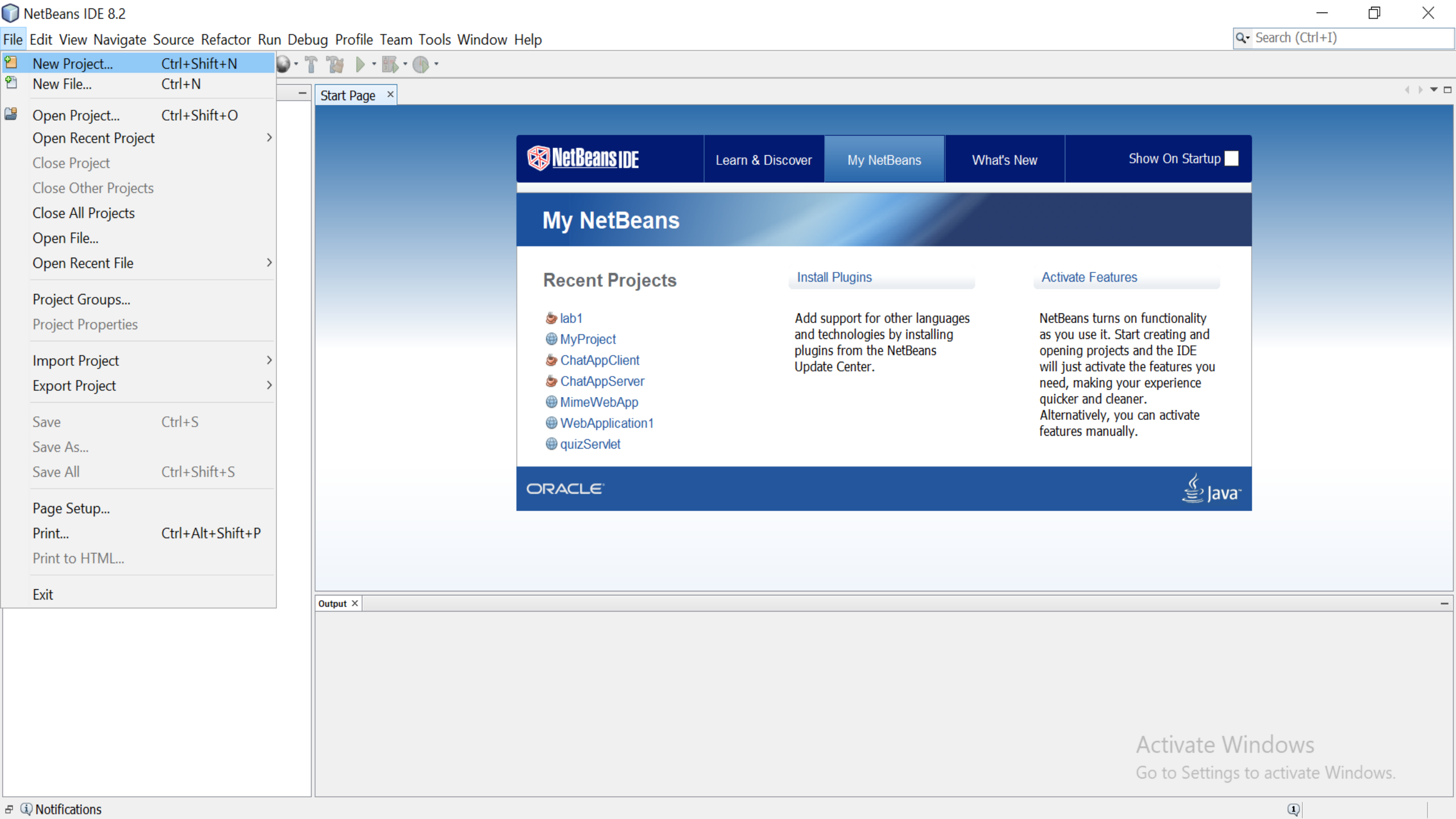
Course Teacher

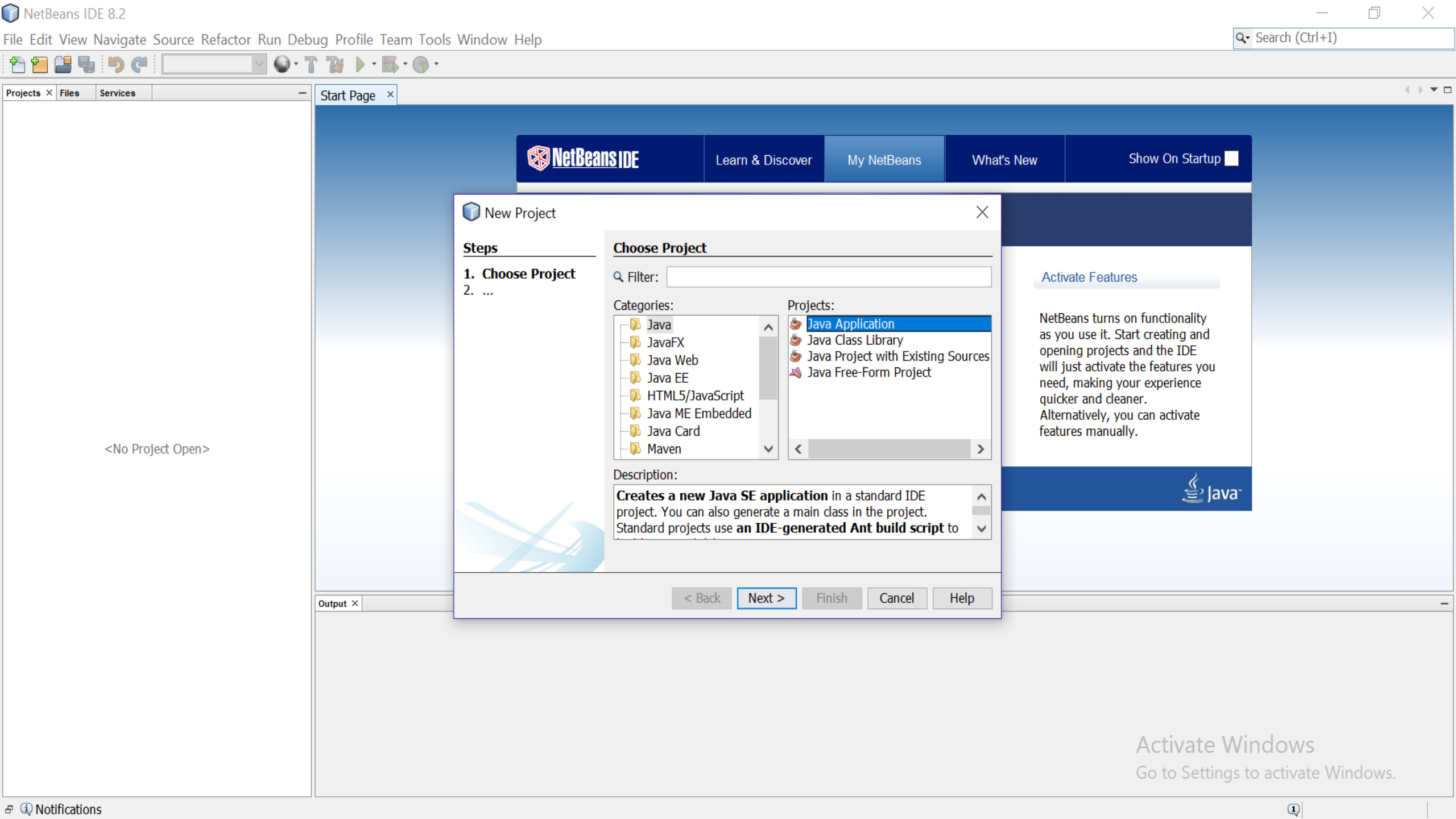
Dr. Shahriar Mahbub, Professor
Nibir Chandra Mandal, Lecturer
Nowshin Nawar Arony, Lecturer

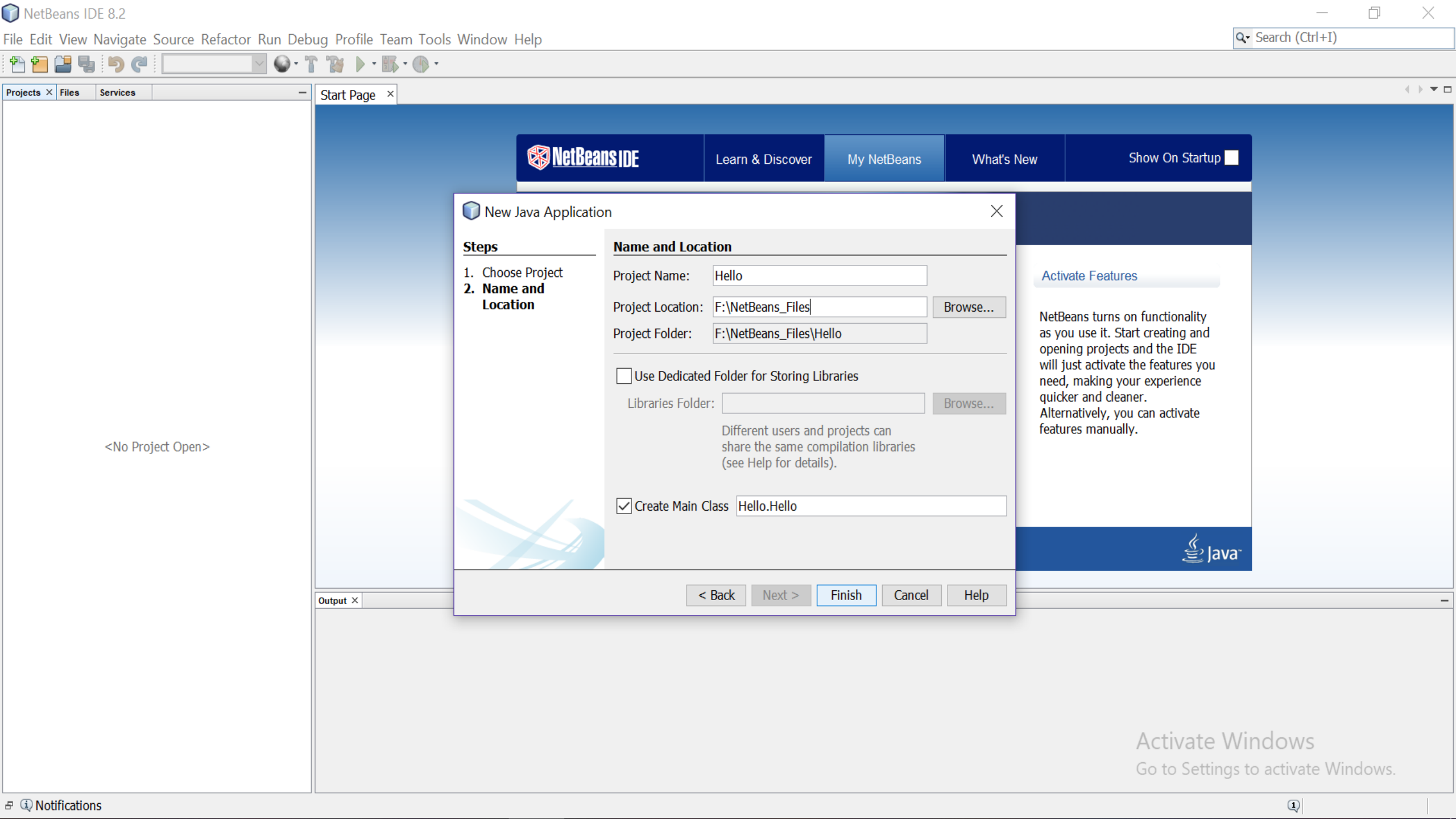
Using Netbeans IDE

Download link:
https://netbeans.org/downloads/8.2/


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Files
Services
<No Project Open>
Start Page
NetBeans IDE
Learn & Discover
My NetBeans
What's New
Show On Startup
My NetBeans
Recent Projects
lab1
MyProject
ChatAppClient
ChatAppServer
MimeWebApp
WebApplication1
quizServlet
Install Plugins
Add support for other languages and technologies by installing plugins from the NetBeans Update Center.
Activate Features
NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.
ORACLE
Java
Output
Activate Windows
Go to Settings to activate Windows.
Notifications


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project
Close Project
Close Other Projects
Close All Projects
Open File...
Open Recent File
Project Groups...
Project Properties
Import Project
Export Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Page Setup...
Print... Ctrl+Alt+Shift+P
Print to HTML...
Exit
Start Page
NetBeans IDE
Learn & Discover
My NetBeans
What's New
Show On Startup
My NetBeans
Recent Projects
lab1
MyProject
ChatAppClient
ChatAppServer
MimeWebApp
WebApplication1
quizServlet
Install Plugins
Add support for other languages and technologies by installing plugins from the NetBeans Update Center.
Activate Features
NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.
ORACLE
Java
Output
Activate Windows
Go to Settings to activate Windows.
Notifications


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Files
Services
<No Project Open>
Start Page
NetBeans IDE
Learn & Discover
My NetBeans
What's New
Show On Startup
Activate Features
NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner.
Alternatively, you can activate features manually.
Java
New Project
Steps
1. Choose Project
2. ...
Choose Project
Filter:
Categories:
Java
JavaFX
Java Web
Java EE
HTML5/JavaScript
Java ME Embedded
Java Card
Maven
Projects:
Java Application
Java Class Library
Java Project with Existing Sources
Java Free-Form Project
Description:
Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use an IDE-generated Ant build script to
< Back
Next >
Finish
Cancel
Help
Output
Activate Windows
Go to Settings to activate Windows.
Notifications


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Files
Services
<No Project Open>
Start Page
NetBeans IDE
Learn & Discover
My NetBeans
What's New
Show On Startup
New Java Application
Steps
1. Choose Project
2. Name and Location
Name and Location
Project Name: Hello
Project Location: F:\NetBeans_Files
Browse...
Project Folder: F:\NetBeans_Files\Hello
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Browse...
Different users and projects can share the same compilation libraries (see Help for details).
Create Main Class Hello.Hello
< Back
Next >
Finish
Cancel
Help
Activate Features
NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.
Java
Output
Activate Windows
Go to Settings to activate Windows.
Notifications

Hello - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

<default confi...

Projects Files Services

- Hello
  - Source Packages
    - Hello
      - Hello.java
  - Libraries

Start Page | Hello.java

Source History

```
1  package Hello;
2
3  public class Hello {
4
5
6      public static void main(String[] args) {
7
8          System.out.println("Hello Java");
9
10     }
11
12 }
13
```

Hello.Hello > main

main - Navigator

Members <empty>

- Hello
  - main(String[] args)

Output - Hello (run)

```
run:
Hello Java
BUILD SUCCESSFUL (total time: 0 seconds)
```


Notifications

8:42 INS

Class and Object

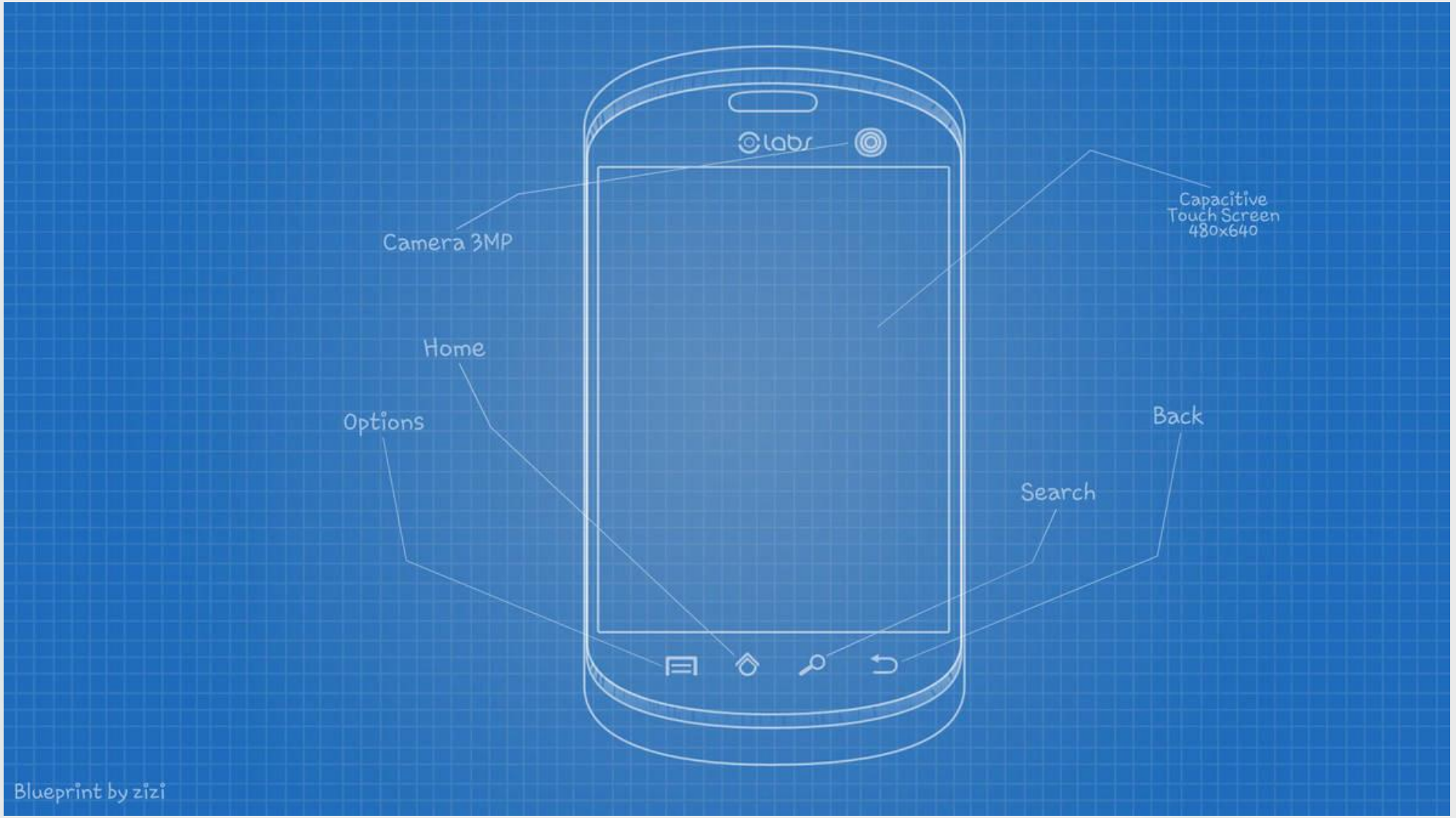
A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

An object is an instance of a class. A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

Object Definitions:

- An object is *a real-world entity*.
- An object is *a runtime entity*.
- The object is *an entity which has state and behavior*.
- The object is *an instance of a class*.

Class In Java



Java Class and Object

Class: SmartPhone

Object-1: iphone

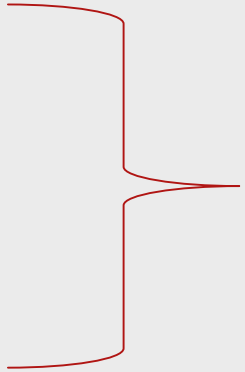
Object-2 : Samsung phone

Object-3: Microsoft phone

Class

A basic smartphone has states or features like:

- **Makers**
- **Operating System**
- **Storage**
- **Screen Size**
- **Model**



Fields

A basic smartphone has behaviors or functions like:

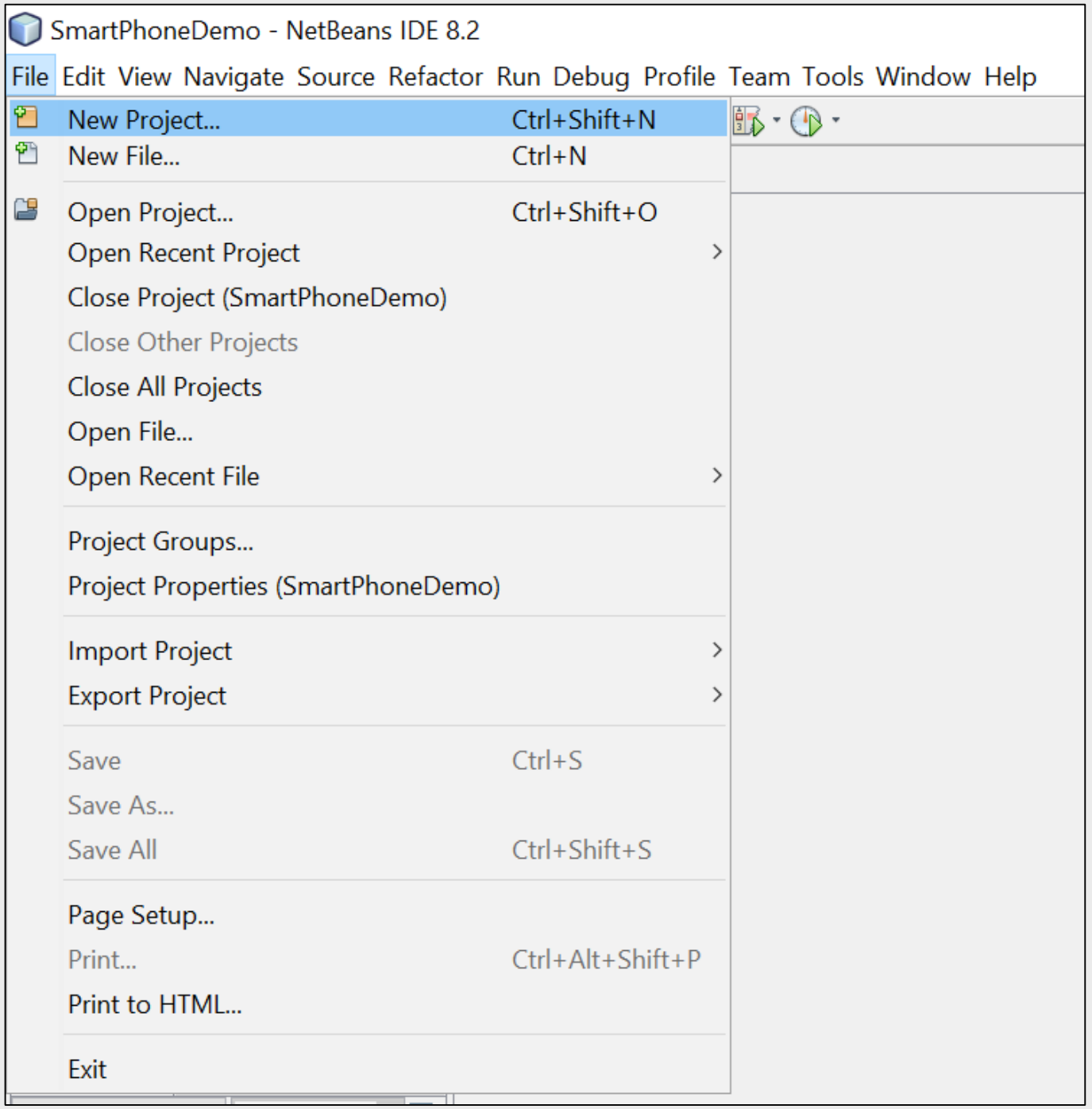
- **Power Duration**
- **Camera**
- **Interface**

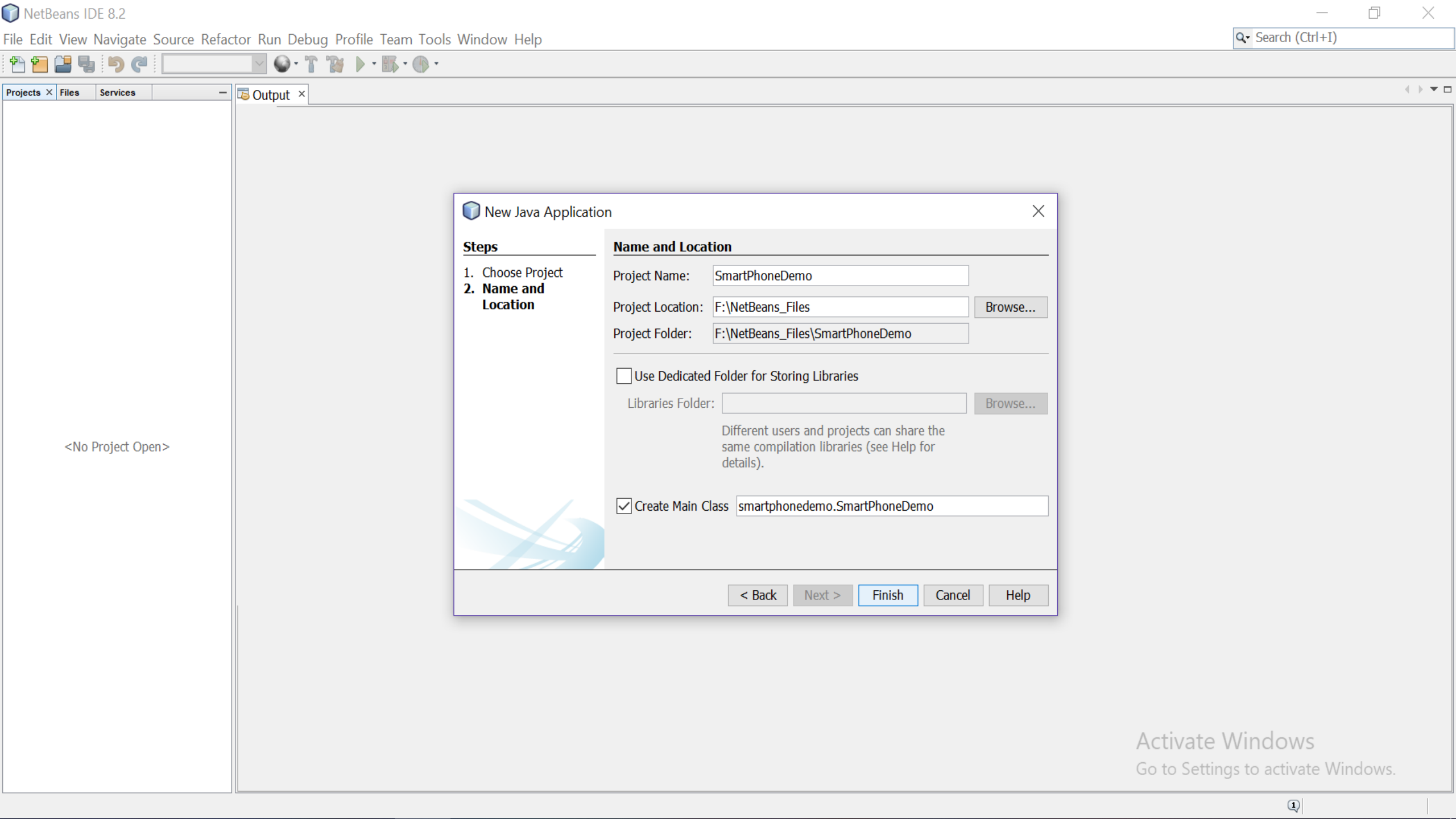
Each phone has its own features. So a class contains basic features and behaviors.

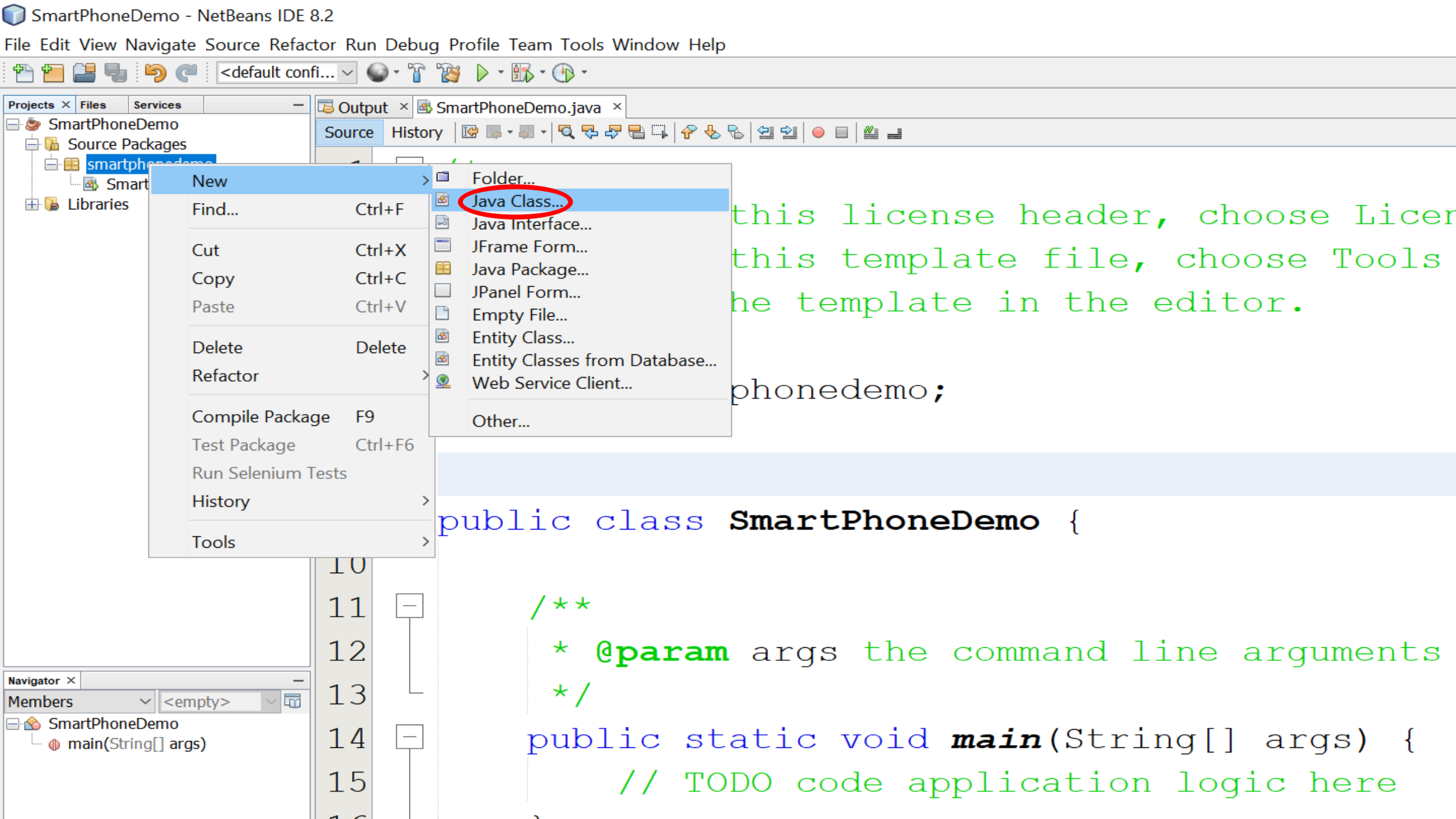
Java Class and Objects

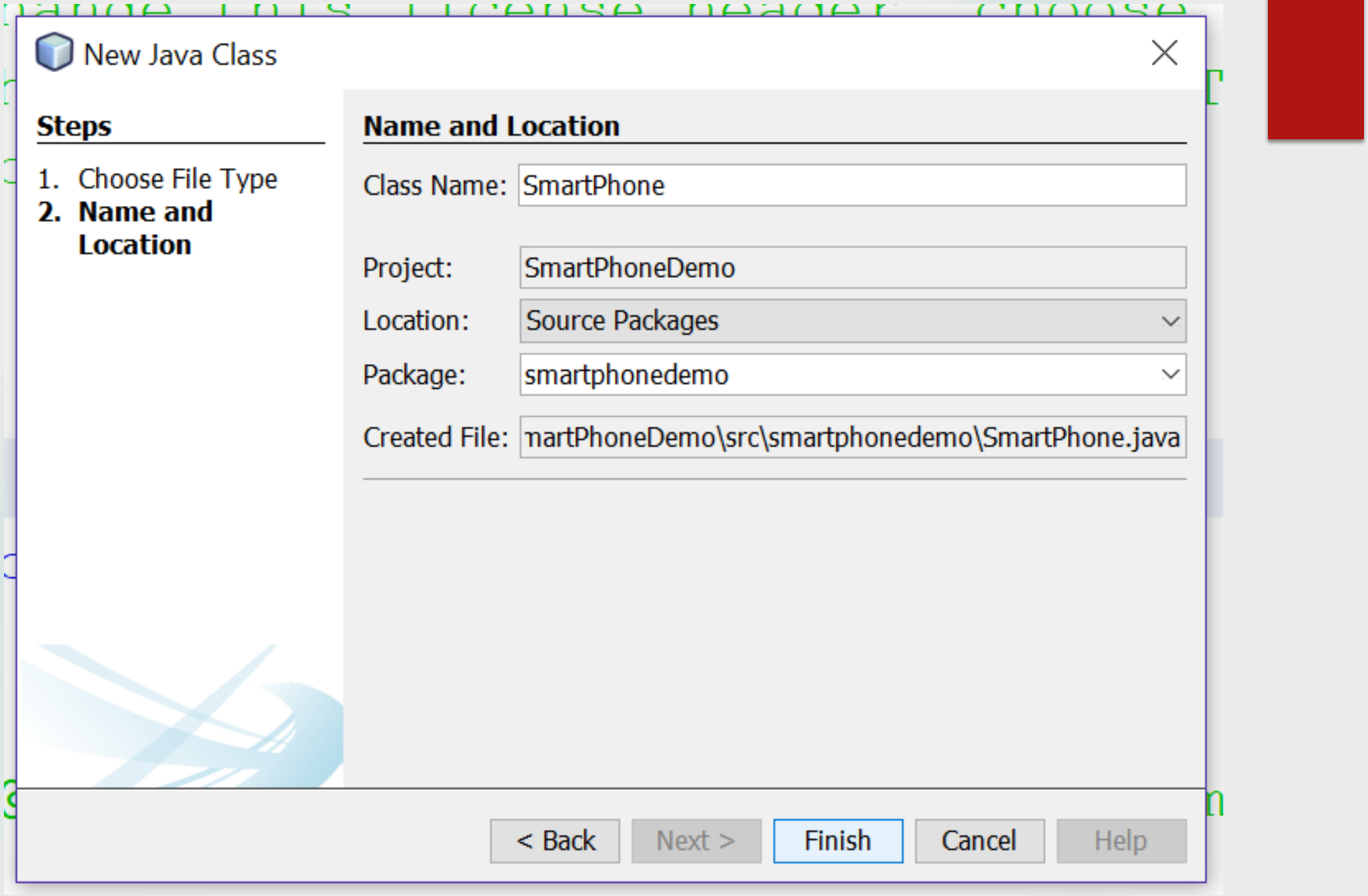
- **A class is a new data type like int, double etc.**
- **This data type is used to create objects.**
- **So a class is a template for an object and an object is an instance of a class.**

Create a New Project




NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Files
Services
Output
<No Project Open>
New Java Application
Steps
1. Choose Project
2. Name and Location
Name and Location
Project Name: SmartPhoneDemo
Project Location: F:\NetBeans_Files
Browse...
Project Folder: F:\NetBeans_Files\SmartPhoneDemo
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Browse...
Different users and projects can share the same compilation libraries (see Help for details).
Create Main Class smartphonedemo.SmartPhoneDemo
< Back
Next >
Finish
Cancel
Help
Activate Windows
Go to Settings to activate Windows.


SmartPhoneDemo - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default confi...
Projects
Files
Services
SmartPhoneDemo
Source Packages
Libraries
New
Find... Ctrl+F
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Refactor
Compile Package F9
Test Package Ctrl+F6
Run Selenium Tests
History
Tools
Folder...
Java Class...
Java Interface...
JFrame Form...
Java Package...
JPanel Form...
Empty File...
Entity Class...
Entity Classes from Database...
Web Service Client...
Other...
Output
SmartPhoneDemo.java
Source
History
this license header, choose Licen
this template file, choose Tools
he template in the editor.
phonedemo;
public class SmartPhoneDemo {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here
Navigator
Members
<empty>
SmartPhoneDemo
main(String[] args)


New Java Class
Steps
1. Choose File Type
2. Name and Location
Name and Location
Class Name: SmartPhone
Project: SmartPhoneDemo
Location: Source Packages
Package: smartphonedemo
Created File: nartPhoneDemo\src\smartphonedemo\SmartPhone.java
< Back
Next >
Finish
Cancel
Help

SmartPhoneDemo - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

<default confi...

Projects Files Services

- SmartPhoneDemo
  - Source Packages
    - smartphonedemo
      - SmartPhone.java
      - SmartPhoneDemo.java
  - Test Packages
  - Libraries
  - Test Libraries

Navigator

Members <empty>

- SmartPhone
  - makers : String
  - modelNum : int
  - os : String

Output SmartPhone.java

Source History

```
1  package smartphonedemo;
2
3  public class SmartPhone {
4
5      public String makers;
6      public String os;
7      public int modelNum;
8
9  }
10
```

10:1 INS

```java
public class SmartPhoneDemo {

    public static void main(String[] args) {

        SmartPhone iphoneObj= new SmartPhone();
        iphoneObj.makers="Apple";
        iphoneObj.os="Mac";
        iphoneObj.modelNum=8;

        System.out.println("Maker = " + iphoneObj.makers);
        System.out.println("OS = " + iphoneObj.os);
        System.out.println("modelNum = " + iphoneObj.modelNum);

    }
}
```

Introducing Methods

```
public class SmartPhone {

    public String makers;
    public String os;
    public int modelNum;

    void setVariables(String m, String op, int modNum)
    {
        makers=m;
        os=op;
        modelNum= modNum;
    }

}
```

```java
public class SmartPhoneDemo {

    public static void main(String[] args) {

        SmartPhone iphoneObj= new SmartPhone();

        iphoneObj.setVariables("Apple", "Mac", 8);

        System.out.println("Maker = " + iphoneObj.makers);
        System.out.println("OS = " + iphoneObj.os);
        System.out.println("modelNum = " + iphoneObj.modelNum);

    }
}
```

Constructors

- **Helps to initialize objects when they are created.**
- **A constructor has the same name as the class.**
- **Looks a method but has no return type. Not even void.**

Constructor Overloading

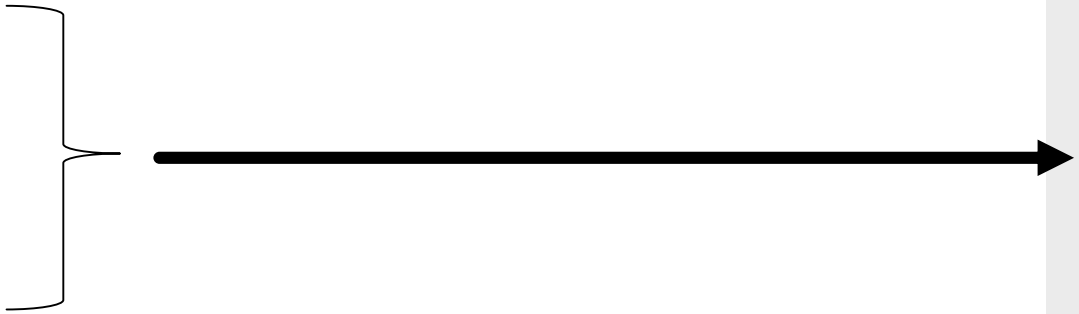
- **creating constructors multiple times with different parameters.**

```
smartphone(){
}

smartphone(double Storage, String Model){
}

smartphone(double Storage, String Model, double screenSize){
}
```

```java
public class SmartPhone {

    public String makers;
    public String os;
    public int modelNum;

    SmartPhone(String m, String op, int modNum)
    {
        makers=m;
        os=op;
        modelNum= modNum;
    }
```

```
public class SmartPhoneDemo {

    public static void main(String[] args) {

        SmartPhone iphoneObj= new SmartPhone("Apple", "Mac", 8);

        //iphoneObj.setVariables("Apple", "Mac", 8);

        System.out.println("Maker = " + iphoneObj.makers);
        System.out.println("OS = " + iphoneObj.os);
        System.out.println("modelNum = " + iphoneObj.modelNum);

    }
```

```java
public class SmartPhone {

    public String makers;
    public String os;
    public int modelNum;

    SmartPhone()
    {


    }
    SmartPhone(String m, String op, int modNum)
    {
        makers=m;
        os=op;
        modelNum= modNum;
    }
```



Default Constructor

this Keyword

```java
public class SmartPhone {

    public String makers;
    public String os;
    public int modelNum;

    SmartPhone(String makers, String os, int modelNum)
    {
        this.makers=makers;
        this.os=os;
        this.modelNum= modelNum;
    }
```

Try yourself

- **Class Name:** Box
- **Variables :** double length, double width, double height
- Initialize using a constructor
- **Create a method:** double getVolume() which returns the volume of the Box = length*width*height

- Go to main method, create an object of Box class, assign values using constructor and print the volume.