Object-Oriented Programming with Java

Assoc. Prof. Dr. Marenglen Biba

General Info

- Course: Object-Oriented Programming with Java (4 credit hours)
- Instructor : Assoc. Prof. Dr. Marenglen Biba
- Office : Faculty building , 2nd floor.
- Office Hours : Tuesday 11-13 or by appointment
- Phone : 0445 12345 / ext. 346
- E-mail : marenglenbiba@unyt.edu.al
- Course page : http://www.marenglenbiba.net/java/

- **Course Location and Time:** Check UNYT schedule
- **Prerequisite:** Introduction to Programming

Course Description

- This course introduces object-oriented programming.
- This course introduces Java language and architecture.
- Students will learn how to program in Java and use some of its most important APIs.
- Special importance will be assigned to the Object-Oriented nature of Java and its use of polymorphism.

Course Outcomes

At the end of the course the student will be able to:

- use Java programming language in object-oriented program design
- understand the Java architecture and use the Java APIs
- understand and use inheritance and polymorphism as implemented in Java
- understand and use the exception handling mechanism of Java
- perform standard input-output operations
- understand and use GUI components

Required Readings

- Java: How to Program. 9th ed. by Deitel & Deitel, (**required**)

- Thinking in Java. 4th ed. by Bruce Eckel, Pearson Education. (**recommended**)

Course Content

- Introduction
- Classes and Objects
- Control Statements
- Arrays and Enumerations
- Inheritance
- Polymorphism
- Collections
- Exceptions
- Standard IO
- GUI components in Java

Grading Policy

Project	40%
Midterm	30%
Final	30%

Deadlines for project submission:

- -20% for the first day of delay,
- -10% for each day of delay after the first.

Technology Expectations

- Internet use is necessary since students should regularly check the course home page
- Continued and regular use of e-mail is expected
  - In all your communications please use:
    - Java course: Student Name Surname
  - In all you communications regarding Projects
    - Java Project : Student Name Surname

- Students must keep copies of all projects sent by e-mail.

Contacts

- How do you contact me?
  - Email
  - In some urgent cases by phone

- How do I contact you?
  - Email
  - In some urgent cases by phone

- I do not have your emails yet!
  - I will use your UNYT e-mail by default.
  - No other e-mail addresses will be used.

Before we start: why failure happens

- Reasons
  - Lack of concentration?
  - Lack of continuity?
  - Lack of determination?
  - Lack of target?
  - Lack of work?
  - …
- Response
  - Hard work will help!
  - Learn by having fun!

Recommendations

- Start studying now
- Do not be shy! Ask any questions that you might have. Every questions makes you a good candidate.
- The professor is a container of knowledge and the goal is to get most of him, thus come and talk.
- Respect the deadlines and the appointments
- Try to study from more than one source, Internet is great!
- If you have any problems come and talk with me in advance so that we can find an appropriate solution

GOOD LUCK!

Lesson 1
Introduction

Outline

- Introduction
- Machine Languages, Assembly Languages and High-Level Languages
- History of C and C++
- History of Java
- Java Class Libraries
- Fortran, COBOL, Pascal and Ada
- BASIC, Visual Basic, Visual C++, C# and .NET
- Typical Java Development Environment

- Lab Session

Introduction

- The core of the course emphasizes achieving program clarity through the proven techniques of object-oriented programming.

- We will follow a live-code approach — Java features presented in complete working Java programs.

- Java is one of today’s most popular languages for developing software.

Structured and OO Programming

- Over the years, many programmers learned structured programming.

- You’ll learn structured programming and object-oriented programming — the key programming methodology used by programmers today.

- You’ll create and work with many software objects.
  - Their internal structure is often built using structured-programming techniques.

Introduction (Cont.)

- Java has become the language of choice for implementing Internet-based applications and software for devices that communicate over a network.

- There are now billions of Java-enabled mobile phones and handheld devices.

- Java is the preferred language for meeting many organizations’ enterprise-wide programming needs.

Personal, Distributed and Client/Server Computing (Cont.)

- Personal computers now are as powerful as the million-dollar machines of just a few decades ago.

- Servers store data that may be used by client computers distributed throughout the network—hence the term client/server computing.

- Java is widely used for writing software for computer networking and for distributed client/server applications.

The Internet and the World Wide Web

- The Internet
  - Global network of computers
  - Has its roots in the 1960s, when funding was supplied by the U.S. Department of Defense.
  - Now accessible by billions of computers and computer-controlled devices worldwide.
- World Wide Web
  - Allows computer users to locate and view multimedia-based documents on almost any subject over the Internet.
- Java is widely used to build programs for WWW

Java Editions

- This course is based on Java Standard Edition (Java SE)
    - JDK: Java Development Kit.
    - http://www.oracle.com/technetwork/java/javase/downloads/index.html

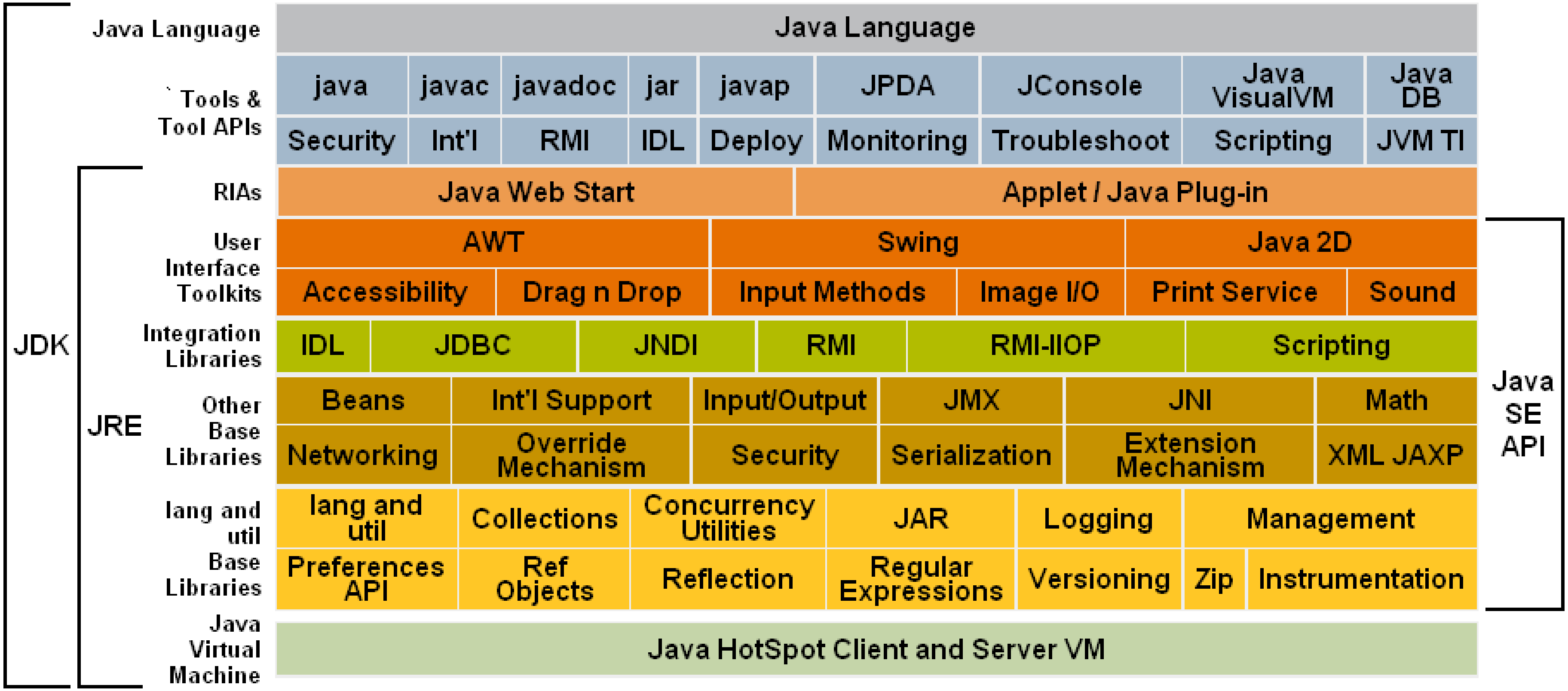
- Java Enterprise Edition (Java EE)
    - geared toward developing large-scale, distributed networking applications and web-based applications.

Java Editions

- Java Micro Edition (Java ME)
  - geared toward developing applications for small, memory-constrained devices, such as cell phones, pagers and PDAs.

- JavaFX
  - a software platform for creating and delivering rich Internet applications enables building applications for desktop, browser and mobile phones. TV set-top boxes, gaming consoles, Blu-ray players and other platforms are planned.

JAVA: A general view



Java Language	Java Language
Tools & Tool APIs	java · javac · javadoc · jar · javap · JPDA · JConsole · Java VisualVM · Java DB
	Security · Int'l · RMI · IDL · Deploy · Monitoring · Troubleshoot · Scripting · JVM TI
RIAs	Java Web Start · Applet / Java Plug-in
User Interface Toolkits	AWT · Swing · Java 2D
	Accessibility · Drag n Drop · Input Methods · Image I/O · Print Service · Sound
Integration Libraries	IDL · JDBC · JNDI · RMI · RMI-IIOP · Scripting
Other Base Libraries	Beans · Int'l Support · Input/Output · JMX · JNI · Math
	Networking · Override Mechanism · Security · Serialization · Extension Mechanism · XML JAXP
lang and util Base Libraries	lang and util · Collections · Concurrency Utilities · JAR · Logging · Management
	Preferences API · Ref Objects · Reflection · Regular Expressions · Versioning · Zip · Instrumentation
Java Virtual Machine	Java HotSpot Client and Server VM

Description of Java Conceptual Diagram

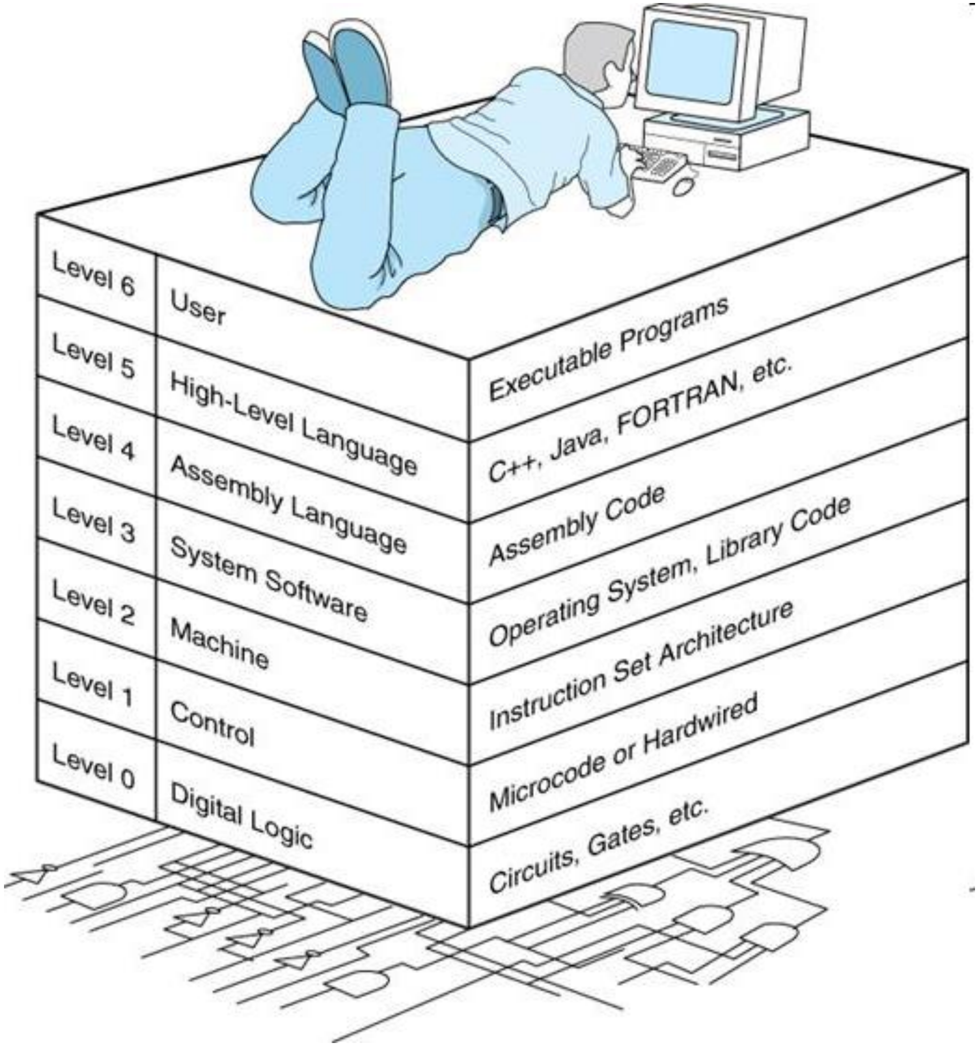
- Full description of APIs
  - http://download.oracle.com/javase/7/docs/technotes/guides/index.html#jre-jdk

Machine Languages, Assembly Languages and High-Level Languages

- Programmers write instructions in various programming languages, some directly understandable by computers and others requiring intermediate translation steps.

- Three general language types:
  - Machine languages
  - Assembly languages
  - High-level languages

Wish you were here!



Machine Languages, Assembly Languages and High-Level Languages (Cont.)

- Any computer can directly understand only its own machine language.
  - This is the computer's "natural language," defined by its hardware design.
  - Generally consist of strings of numbers (ultimately reduced to 1s and 0s) that instruct computers to perform their most elementary operations one at a time.
  - Machine dependent — a particular machine language can be used on only one type of computer.

Machine Languages, Assembly Languages and High-Level Languages (Cont.)

- English-like abbreviations that represent elementary operations formed the basis of assembly languages.

- Translator programs called assemblers convert assembly-language programs to machine language.

Machine Languages, Assembly Languages and High-Level Languages (Cont.)

- High-level languages
  - Single statements accomplish substantial tasks.
  - Compilers convert high-level language programs into machine language.
  - Allow you to write instructions that look almost like everyday English and contain commonly used mathematical notations.
- C, C++, Microsoft’s .NET languages (e.g., Visual Basic, Visual C++ and C#) are among the most widely used high-level programming languages;
- Java is by far the most widely used language.

Machine Languages, Assembly Languages and High-Level Languages (Cont.)

- Compiling a high-level language program into machine language can take a considerable amount of computer time.

- Interpreter programs execute high-level language programs directly, although slower than compiled programs run.

- Java uses a clever mixture of compilation and interpretation to run programs.

History of C and C++

- Java evolved from C++, which evolved from C, which evolved from BCPL and B.
- C
  - Originally implemented in 1972
  - Evolved from B by Dennis Ritchie at Bell Laboratories
  - Became widely known as the UNIX operating system’s development language
  - Today, most of the code for general-purpose operating systems is written in C or C++.

History of C and C++ (Cont.)

- C++
  - An extension of C
  - Developed by Bjarne Stroustrup in the early 1980s at Bell Laboratories
  - Provides capabilities for *object-oriented programming.*
  - Hybrid language — it's possible to program in either a C-like style, an object-oriented style or both.

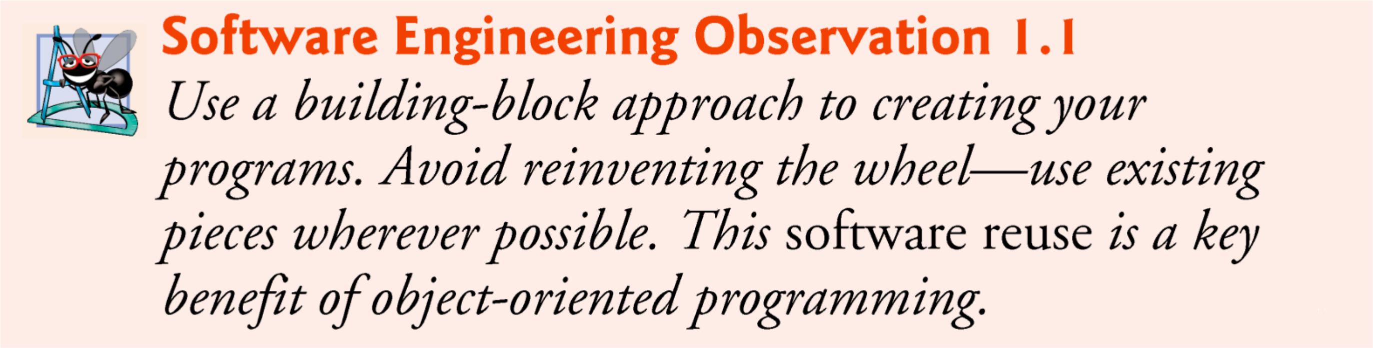
History of Java

- Microprocessors are having a profound impact in intelligent consumer-electronic devices.
- 1991
  - Recognizing this, Sun Microsystems funded an internal corporate research project, which resulted in a C++-based language named Java
  - Created by James Gosling.
- 1993
  - The web exploded in popularity
  - Sun saw the potential of using Java to add dynamic content to web pages.
- Java garnered the attention of the business community because of the phenomenal interest in the web.

Java Class Libraries

- Java programs consist of pieces called classes.
- Classes include methods that perform tasks and return information when the tasks complete.
- Java class libraries
  - Rich collections of existing classes
  - Also known as the Java APIs (Application Programming Interfaces)
- Two aspects to learning the Java "world."
  - The Java language itself
  - The classes in the extensive Java class libraries
- Download the Java API documentation
  - http://www.oracle.com/technetwork/java/javase/documentation/index.html

Avoid reinventing the wheel!



Software Engineering Observation 1.1

Use a building-block approach to creating your programs. Avoid reinventing the wheel—use existing pieces wherever possible. This software reuse *is a key benefit of object-oriented programming.*

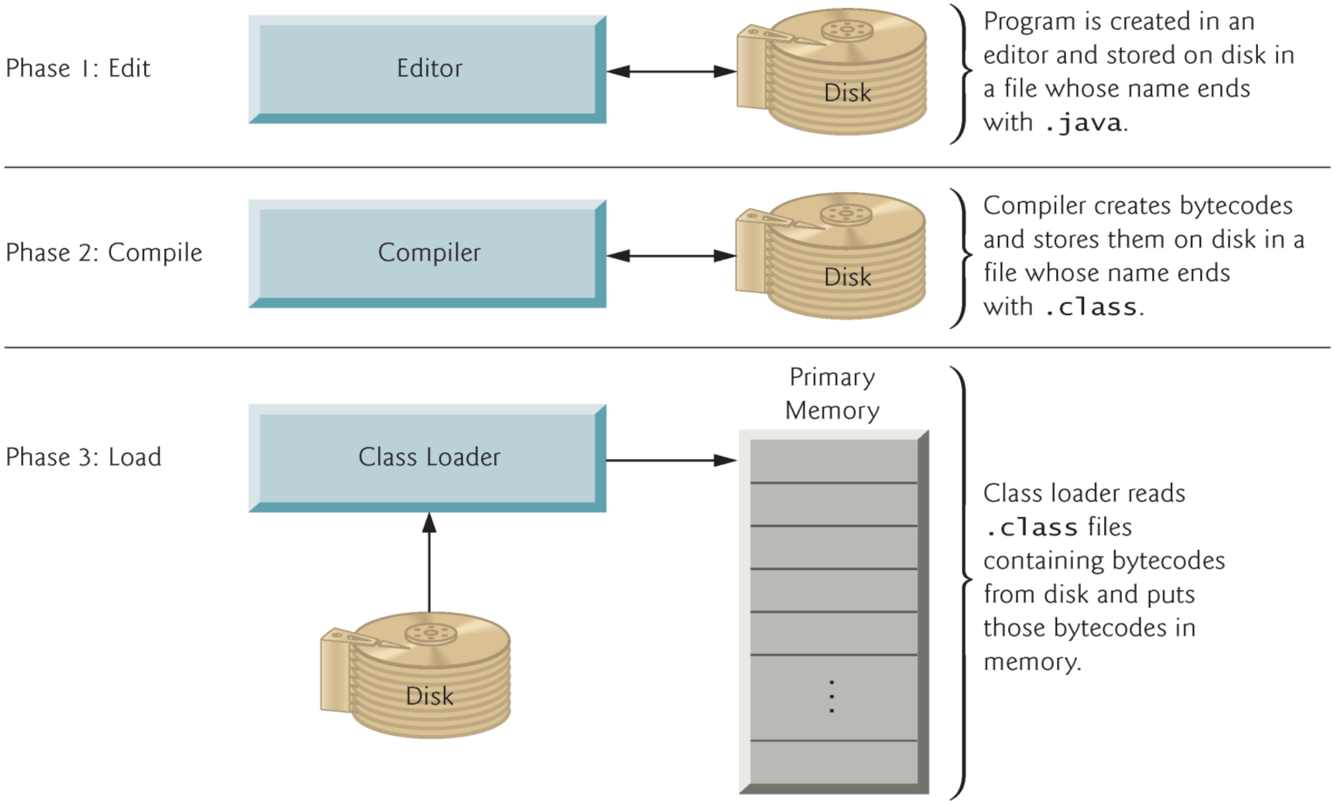
The existing pieces are available Java classes that have been developed before and you can use them for your own programs.

Typical Java Development Environment

- Java program development and execution cycle (illustrated in Fig. 1.1).

Typical Java Development Environment (Cont.)

- Java programs normally go through five phases
  - edit
  - compile
  - load
  - verify
  - execute

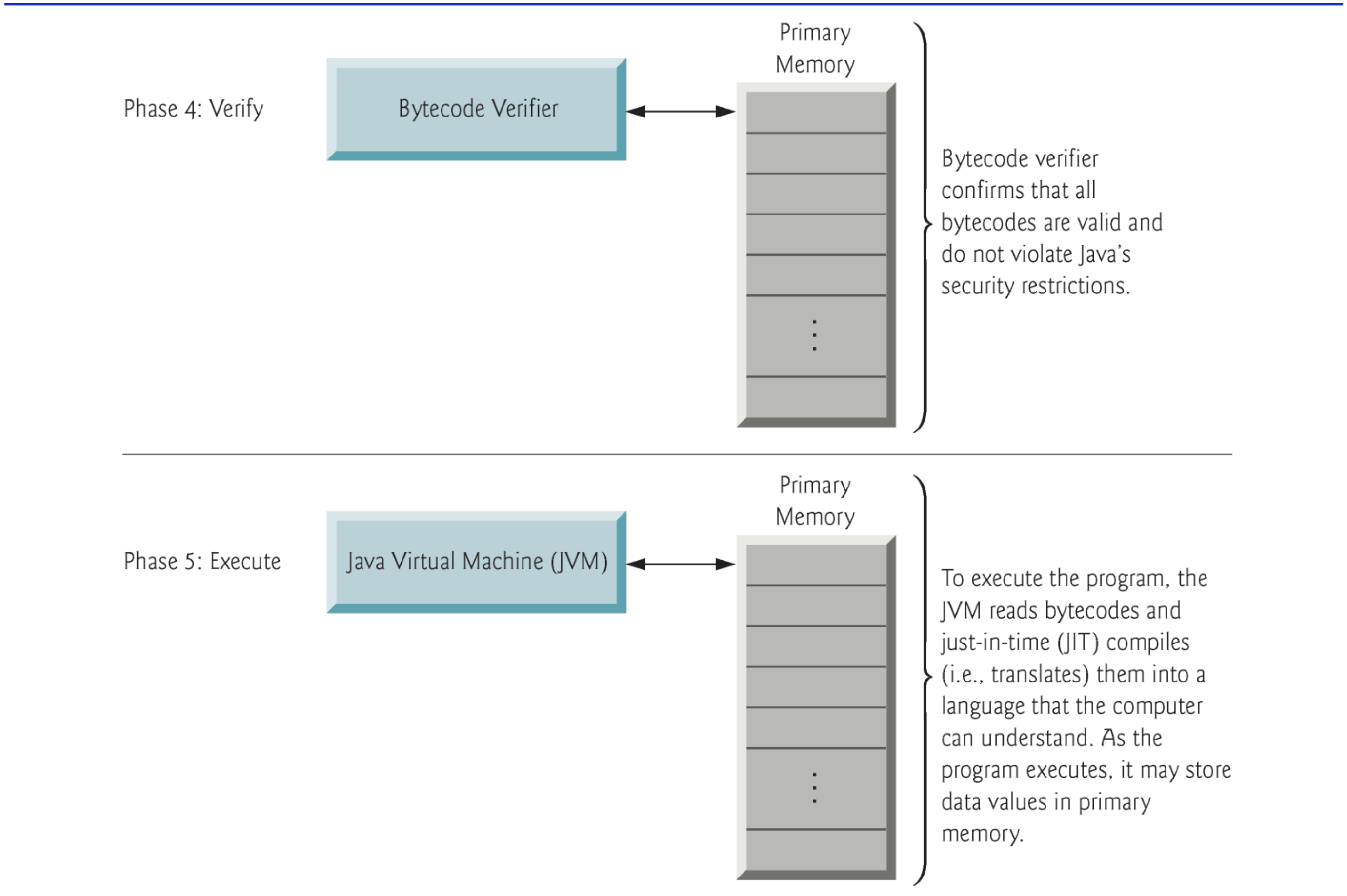


Program is created in an editor and stored on disk in a file whose name ends with `.java`.

Compiler creates bytecodes and stores them on disk in a file whose name ends with `.class`.

Class loader reads `.class` files containing bytecodes from disk and puts those bytecodes in memory.

Fig. 1.1 | Typical Java development environment. (Part 1 of 2.)



Bytecode verifier confirms that all bytecodes are valid and do not violate Java's security restrictions.

To execute the program, the JVM reads bytecodes and just-in-time (JIT) compiles (i.e., translates) them into a language that the computer can understand. As the program executes, it may store data values in primary memory.

Fig. 1.1 | Typical Java development environment. (Part 2 of 2.)

Typical Java Development Environment (Cont.)

- We discuss these phases in the context of the Java SE Development Kit 8 (JDK8) from Sun Microsystems, Inc.
- Download the most up-to-date JDK and its documentation from:
  - http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html
  - Carefully follow the installation instructions for the JDK provided in the Before You Begin section of the book to ensure that you set up your computer properly to compile and execute Java programs.
- Sun's New to Java Center at:
  - http://www.oracle.com/technetwork/java/javase/overview/index.html

Typical Java Development Environment (Cont.)

- Phase 1 consists of editing a file with an editor program *(normally known simply as an editor).*
  - Type a Java program (source code) using the editor
  - Make any necessary corrections
  - Save the program
    - A file name ending with the `.java` extension indicates that the file contains Java source code.
  - Linux editors: `vi` and `emacs`.
  - Windows editors: Notepad, EditPlus (`www.editplus.com`), TextPad (`www.textpad.com`) and jEdit (`www.jedit.org`).

Typical Java Development Environment (Cont.)

- Integrated development environments (IDEs)
  - Provide tools that support the software-development process, including editors for writing and editing programs and debuggers for locating logic errors — errors that cause programs to execute incorrectly.
- Popular IDEs
  - Eclipse (`www.eclipse.org`)
  - NetBeans (`www.netbeans.org`)
  - Oracle JDeveloper
  - JBuilder (`www.codegear.com`)
  - JCreator (`www.jcreator.com`)
  - BlueJ (`www.blueJ.org`)
  - jGRASP (`www.jgrasp.org`)

Typical Java Development Environment (Cont.)

- Phase 2
  - Use the command `javac` (the Java compiler) to compile a program. For example, to compile a program called `Welcome.java`, you’d type

    ```
    javac Welcome.java
    ```

  - If the program compiles, the compiler produces a .class file called `Welcome.class` that contains the compiled version of the program.

Typical Java Development Environment (Cont.)

- Java compiler translates Java source code into bytecodes that represent the tasks to execute.
- Bytecodes are executed by the Java Virtual Machine (JVM)—a part of the JDK and the foundation of the Java platform.
- JRE
  - You can download this from the same page of JDK
- Virtual machine (VM) — a software application that simulates a computer
  - Hides the underlying operating system and hardware from the programs that interact with it.
- If the same VM is implemented on many computer platforms, applications that it executes can be used on all those platforms.

JRE and JDK

- **Sun Microsystems**
  - provides two principal software products in the Java™ Platform, Standard Edition (Java™ SE) family:

- **Java SE Runtime Environment (JRE)**
  - The JRE provides the libraries, Java virtual machine, and other components necessary for you to *run* applications written in the Java programming language.
  - This runtime environment can be redistributed with applications to make them free-standing.

- **Java SE Development Kit (JDK)**
  - The JDK includes the JRE plus command-line development tools such as compilers and debuggers that are necessary or useful for *developing* applets and applications.

JVM

- **Java Virtual Machines**
  - The Java virtual machine is an abstract computing machine that has an instruction set and manipulates memory at run time.
  - The Java virtual machine is ported to different platforms to provide hardware- and operating system-independence.
  - The Java Platform, Standard Edition provides two implementations of the Java virtual machine (VM):

- **Java HotSpot Client VM**
  - The client VM is an implementation for platforms typically used for client applications. The client VM is tuned for reducing start-up time and memory footprint.
- **Java HotSpot Server VM**
  - The server VM is an implementation designed for maximum program execution speed, trading off launch time and memory.

Typical Java Development Environment (Cont.)

- Bytecodes are platform independent
  - They do not depend on a particular hardware platform.
- Bytecodes are portable
  - The same bytecodes can execute on any platform containing a JVM that understands the version of Java in which the bytecodes were compiled.
- The JVM is invoked by the `java` command. For example, to execute a Java application called `Welcome`, you'd type the command

```
java Welcome
```

Typical Java Development Environment (Cont.)

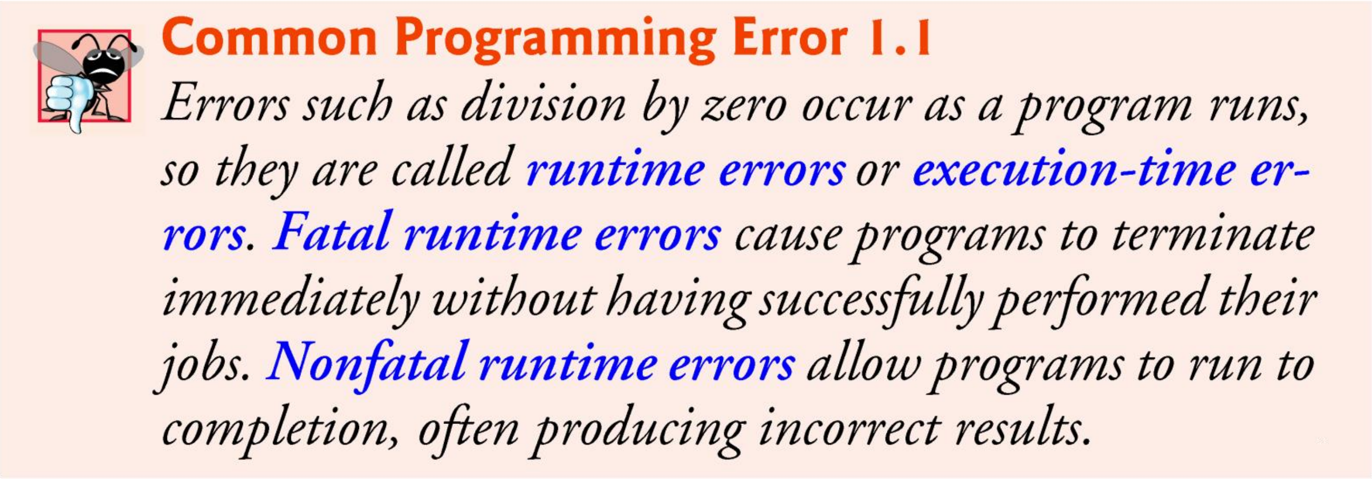
- Phase 3
  - The JVM places the program in memory to execute it
    - This is known as loading.
  - Class loader takes the `.class` files containing the program’s bytecodes and transfers them to primary memory.
  - Also loads any of the `.class` files provided by Java that your program uses.
    - The `.class` files can be loaded from a disk on your system or over a network.

Typical Java Development Environment (Cont.)

- Phase 4
  - As the classes are loaded, the bytecode verifier examines their bytecodes
    - Ensures that they are valid and do not violate Java’s security restrictions.
  - Java enforces strong security to make sure that Java programs arriving over the network do not damage your files or your system (as computer viruses and worms might).

Typical Java Development Environment (Cont.)

- Phase 5
  - The JVM executes the program's bytecodes.
  - JVM typically uses a combination of interpretation and just-in-time (JIT) compilation.
  - Analyzes the bytecodes as they are interpreted, searching for hot spots — parts of the bytecodes that execute frequently.
  - A just-in-time (JIT) compiler (the Java HotSpot compiler) translates the bytecodes into the underlying computer's machine language.
  - When the JVM encounters these compiled parts again, the faster machine-language code executes.



Common Programming Error 1.1

Errors such as division by zero occur as a program runs, so they are called ***runtime errors*** *or* ***execution-time errors****.* ***Fatal runtime errors*** *cause programs to terminate immediately without having successfully performed their jobs.* ***Nonfatal runtime errors*** *allow programs to run to completion, often producing incorrect results.*

Notes about Java and *Java How to Program, Eighth and Ninth Edition*

- Web-based version of the Java API documentation
  - java.sun.com/javase/8/docs/api/index.html
- Download this documentation to your own computer
  - java.sun.com/javase/downloads/
- Additional technical details on many aspects of Java development
  - java.sun.com/reference/docs/index.html
- JDK Programmers Guide
  - http://java.sun.com/javase/8/docs

Laboratory Session

Test-Driving a Java Application

- Checking your setup. Read the *Before You Begin section of the book to confirm that you've set up Java properly on your computer.*

- *Download Java at:*
  - http://www.oracle.com/technetwork/java/javase/downloads/index.html

- *Writing a simple Java Program.*
  - *Welcome*
  - *Perform simple operations*
  - *Print to the console*

- *Compiling the program*
  - *Javac command or*
  - *Netbeans*

- *Executing*
  - *The java command or*
  - *Run in Netbeans*

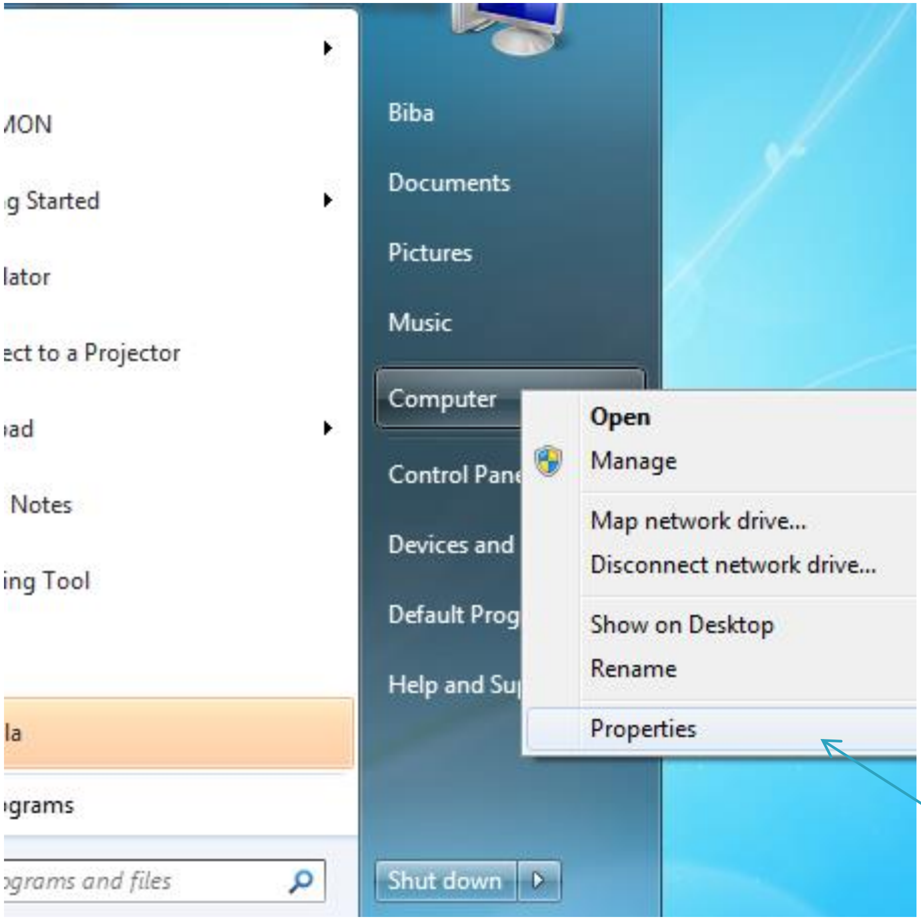
First Program in Java

```
public class Operator {

public static void main( String args[ ] ) {
    System.out.println("*************************************************");
    System.out.println( "Welcome, this is your first program in Java" );
    int x = 30;
    int y = 2;
    int result = x * y + 9 / 3;
    System.out.print("The result is: ");
    System.out.println(result);
    System.out.println("*************************************************");
}
}
```

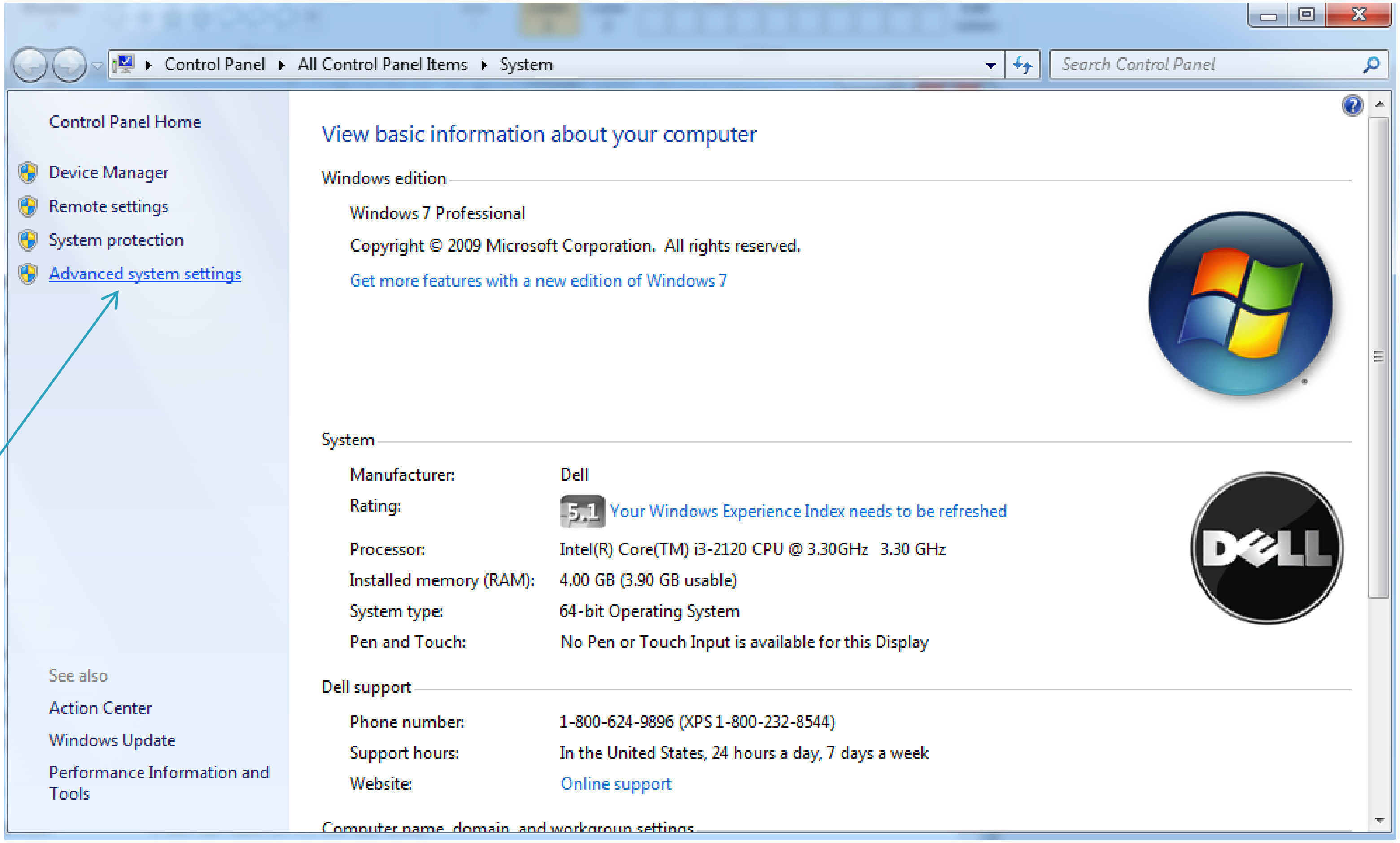
Saving the program

- Save the program
  - A file name ending with the `.java` extension indicates that the file contains Java source code.

Compiling using the command prompt: setting environmental variables



Setting environmental variables



View basic information about your computer

Windows edition

Windows 7 Professional

Copyright © 2009 Microsoft Corporation. All rights reserved.

Get more features with a new edition of Windows 7

System

Manufacturer: Dell

Rating: 5.1 Your Windows Experience Index needs to be refreshed

Processor: Intel(R) Core(TM) i3-2120 CPU @ 3.30GHz 3.30 GHz

Installed memory (RAM): 4.00 GB (3.90 GB usable)

System type: 64-bit Operating System

Pen and Touch: No Pen or Touch Input is available for this Display

Dell support

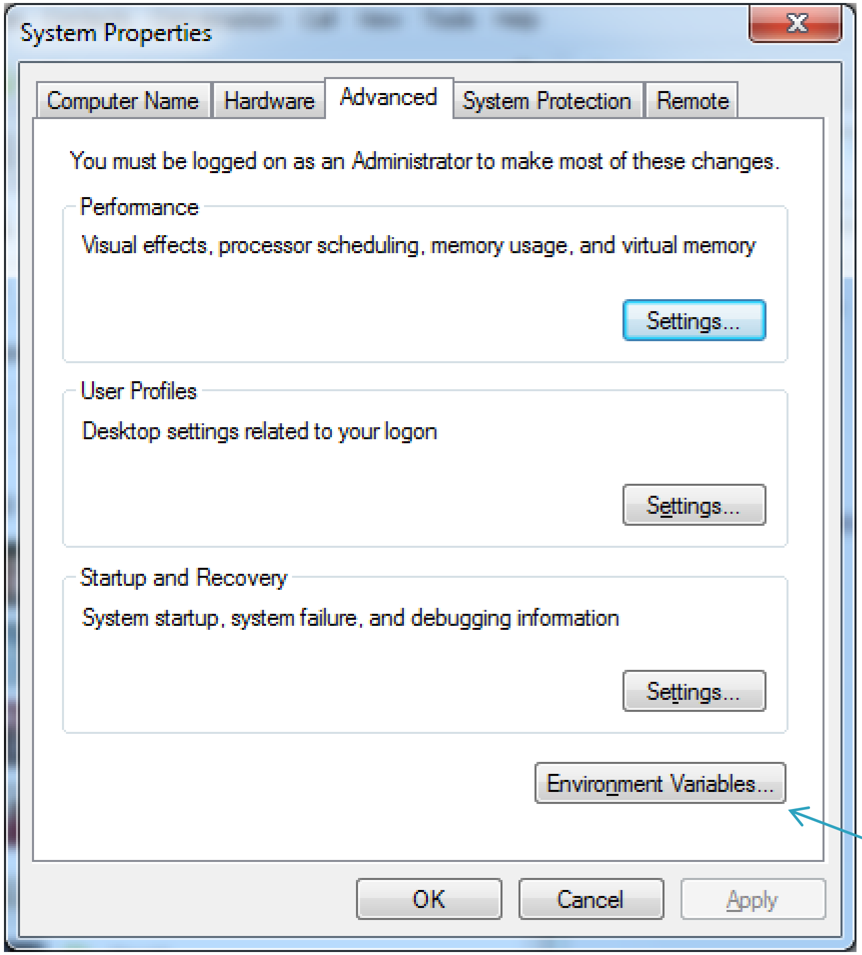
Phone number: 1-800-624-9896 (XPS 1-800-232-8544)

Support hours: In the United States, 24 hours a day, 7 days a week

Website: Online support

Computer name, domain, and workgroup settings

Setting environmental variables



System Properties

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

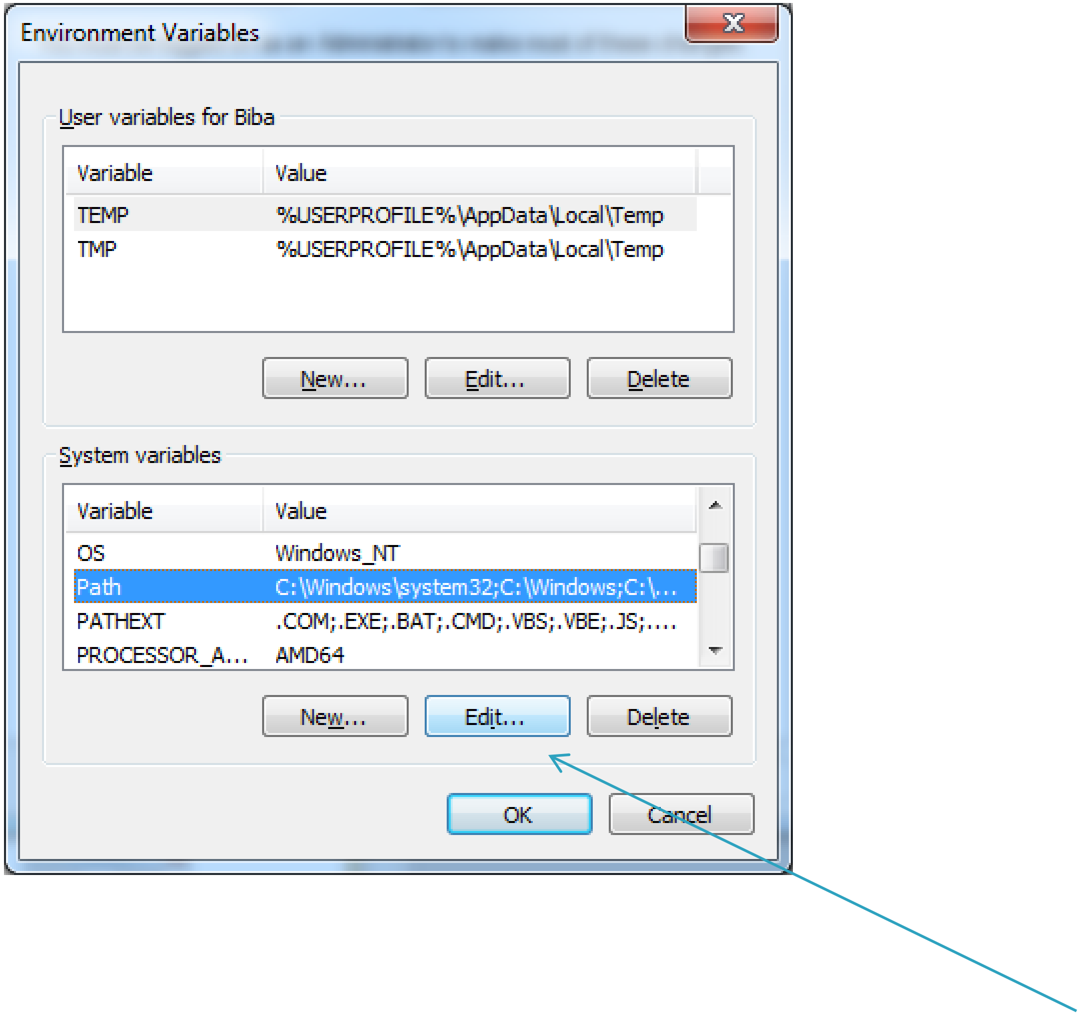
User Profiles

Desktop settings related to your logon

Startup and Recovery

System startup, system failure, and debugging information

Setting environmental variables



Environment Variables

User variables for Biba

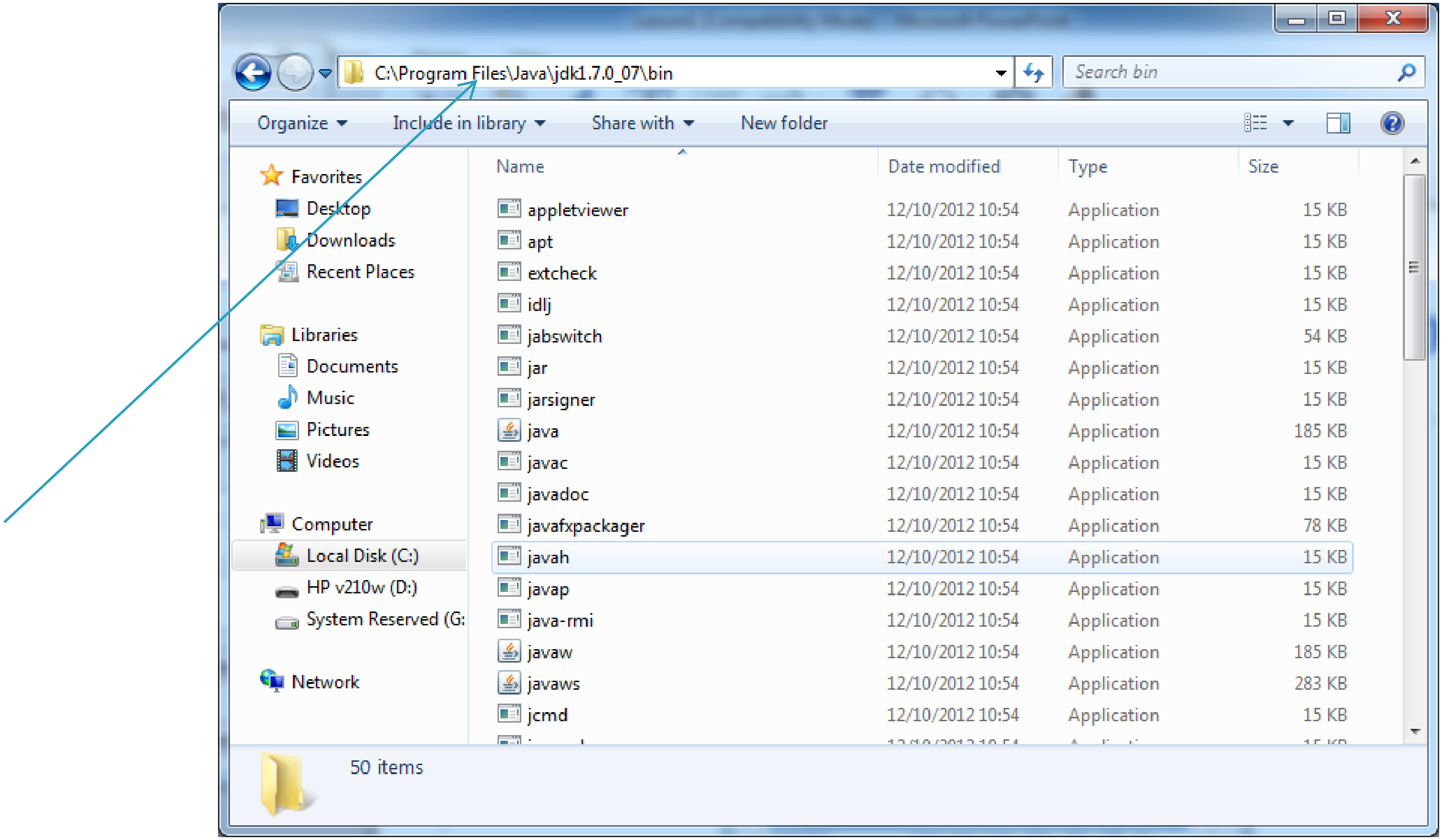
Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64

Setting environmental variables

- Locate Java SDK. Look for JDK folder/bin



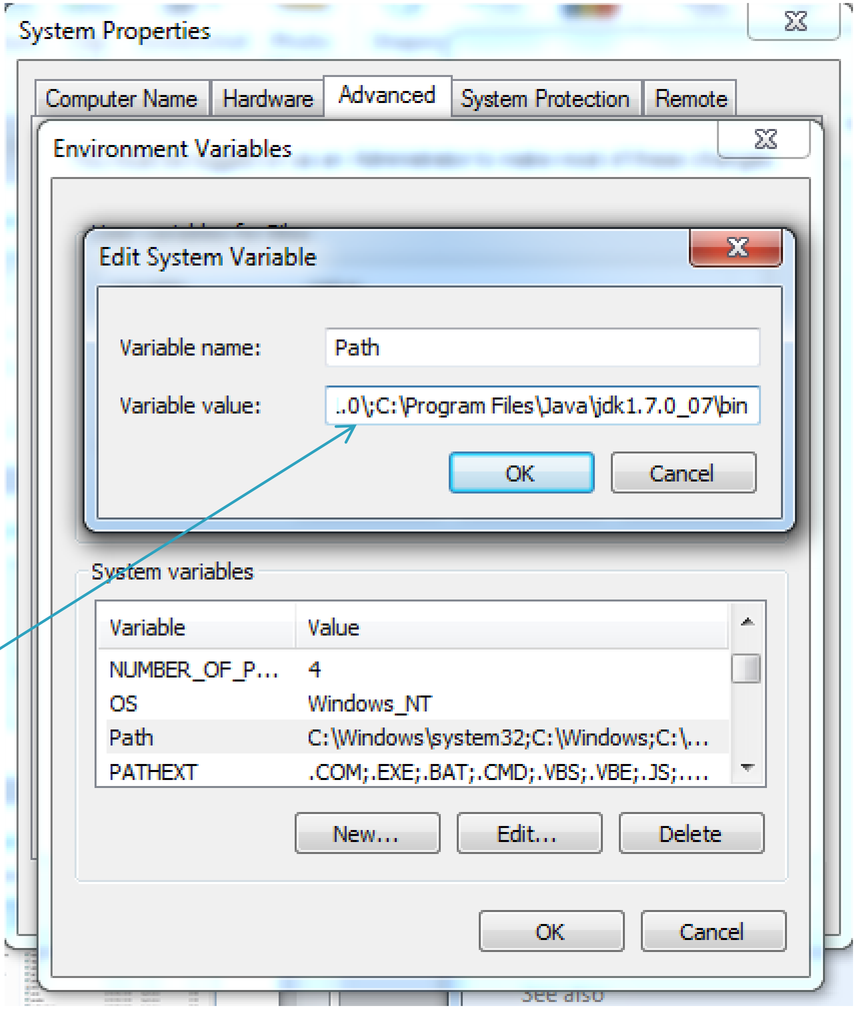
C:\Program Files\Java\jdk1.7.0_07\bin

Name	Date modified	Type	Size
appletviewer	12/10/2012 10:54	Application	15 KB
apt	12/10/2012 10:54	Application	15 KB
extcheck	12/10/2012 10:54	Application	15 KB
idlj	12/10/2012 10:54	Application	15 KB
jabswitch	12/10/2012 10:54	Application	54 KB
jar	12/10/2012 10:54	Application	15 KB
jarsigner	12/10/2012 10:54	Application	15 KB
java	12/10/2012 10:54	Application	185 KB
javac	12/10/2012 10:54	Application	15 KB
javadoc	12/10/2012 10:54	Application	15 KB
javafxpackager	12/10/2012 10:54	Application	78 KB
javah	12/10/2012 10:54	Application	15 KB
javap	12/10/2012 10:54	Application	15 KB
java-rmi	12/10/2012 10:54	Application	15 KB
javaw	12/10/2012 10:54	Application	185 KB
javaws	12/10/2012 10:54	Application	283 KB
jcmd	12/10/2012 10:54	Application	15 KB

50 items

Compiling using the command prompt

- Set the environment variable in Windows. This is done only once.



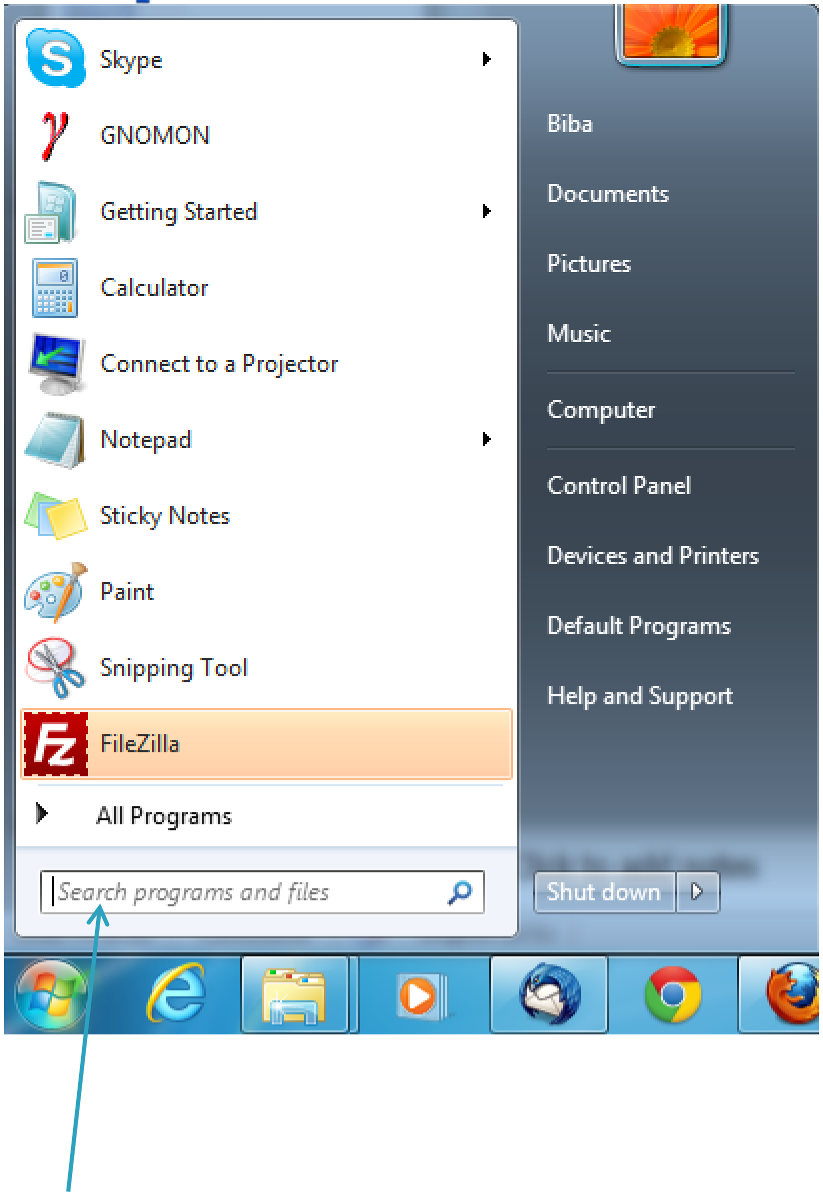
Variable name: Path

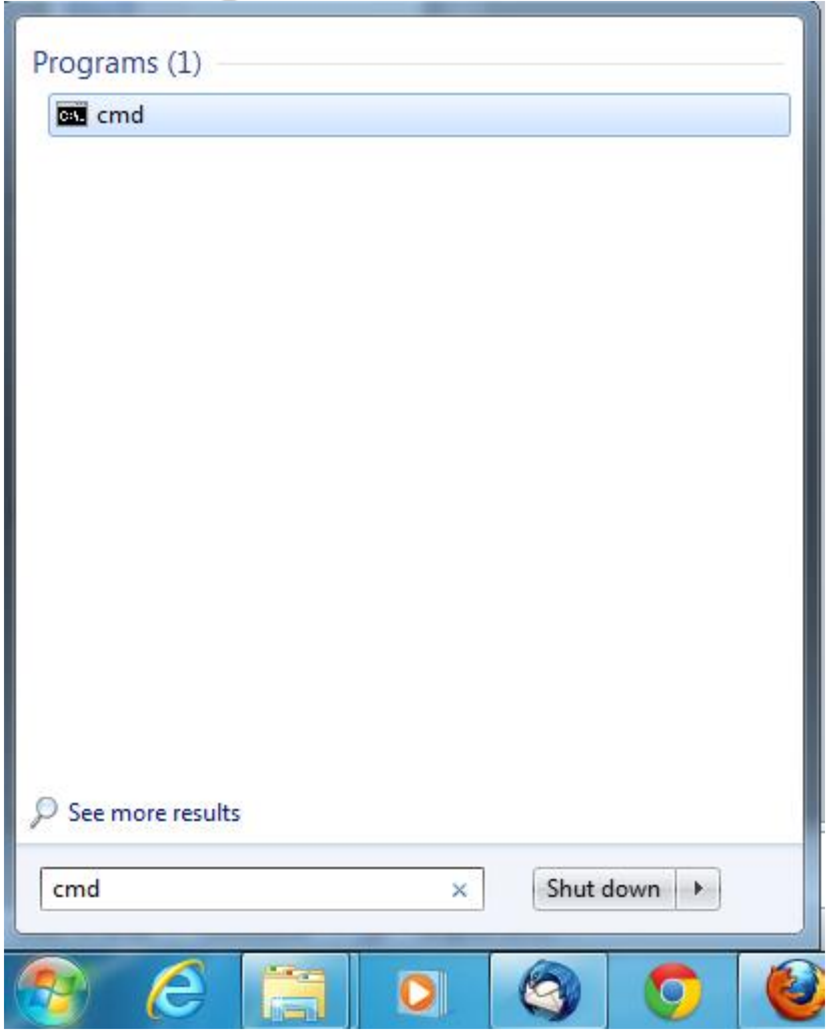
Variable value: ..0\;C:\Program Files\Java\jdk1.7.0_07\bin

Variable	Value
NUMBER_OF_P...	4
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

The jdk/bin path.
Put ";" after every path.

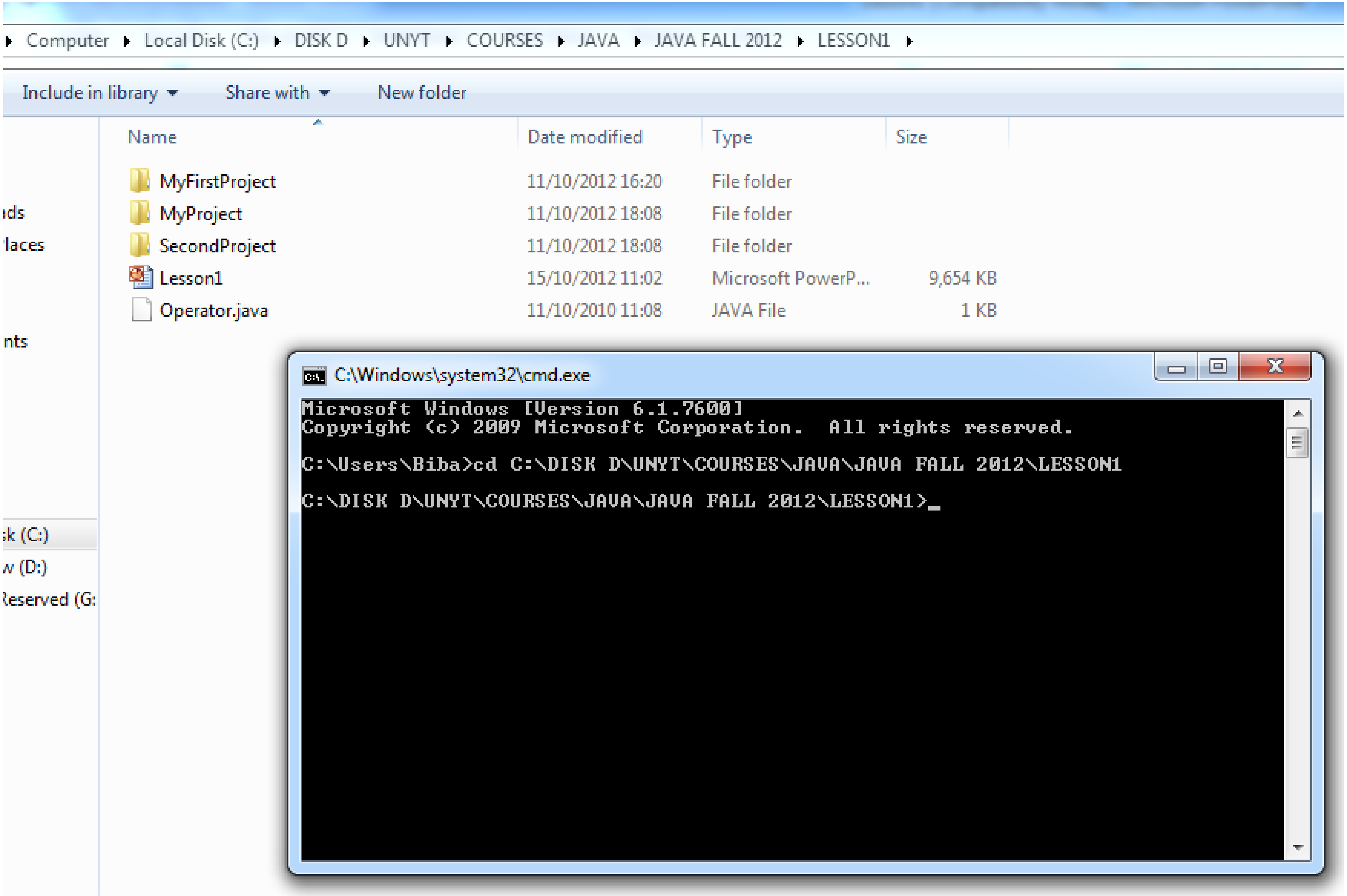
Open a command prompt





cmd

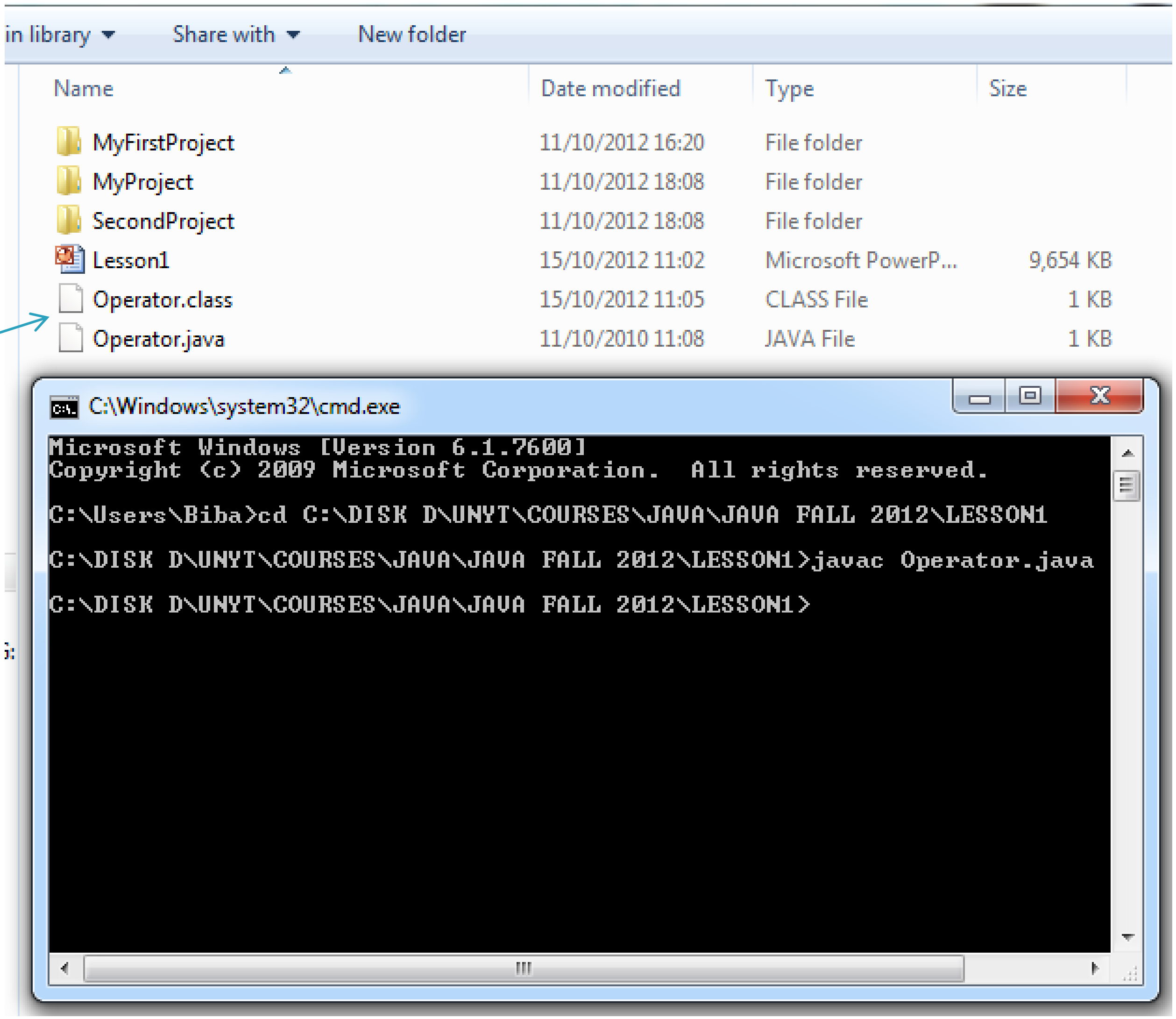
Change directory to the .java file


Computer ▸ Local Disk (C:) ▸ DISK D ▸ UNYT ▸ COURSES ▸ JAVA ▸ JAVA FALL 2012 ▸ LESSON1 ▸
Include in library ▾
Share with ▾
New folder
ıds
laces
nts
:k (C:)
w (D:)
Reserved (G:
C:\Windows\system32\cmd.exe

Name	Date modified	Type	Size
MyFirstProject	11/10/2012 16:20	File folder	
MyProject	11/10/2012 18:08	File folder	
SecondProject	11/10/2012 18:08	File folder	
Lesson1	15/10/2012 11:02	Microsoft PowerP...	9,654 KB
Operator.java	11/10/2010 11:08	JAVA File	1 KB

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

C:\Users\Biba>cd C:\DISK D\UNYT\COURSES\JAVA\JAVA FALL 2012\LESSON1

C:\DISK D\UNYT\COURSES\JAVA\JAVA FALL 2012\LESSON1>_
```

Compile with javac command

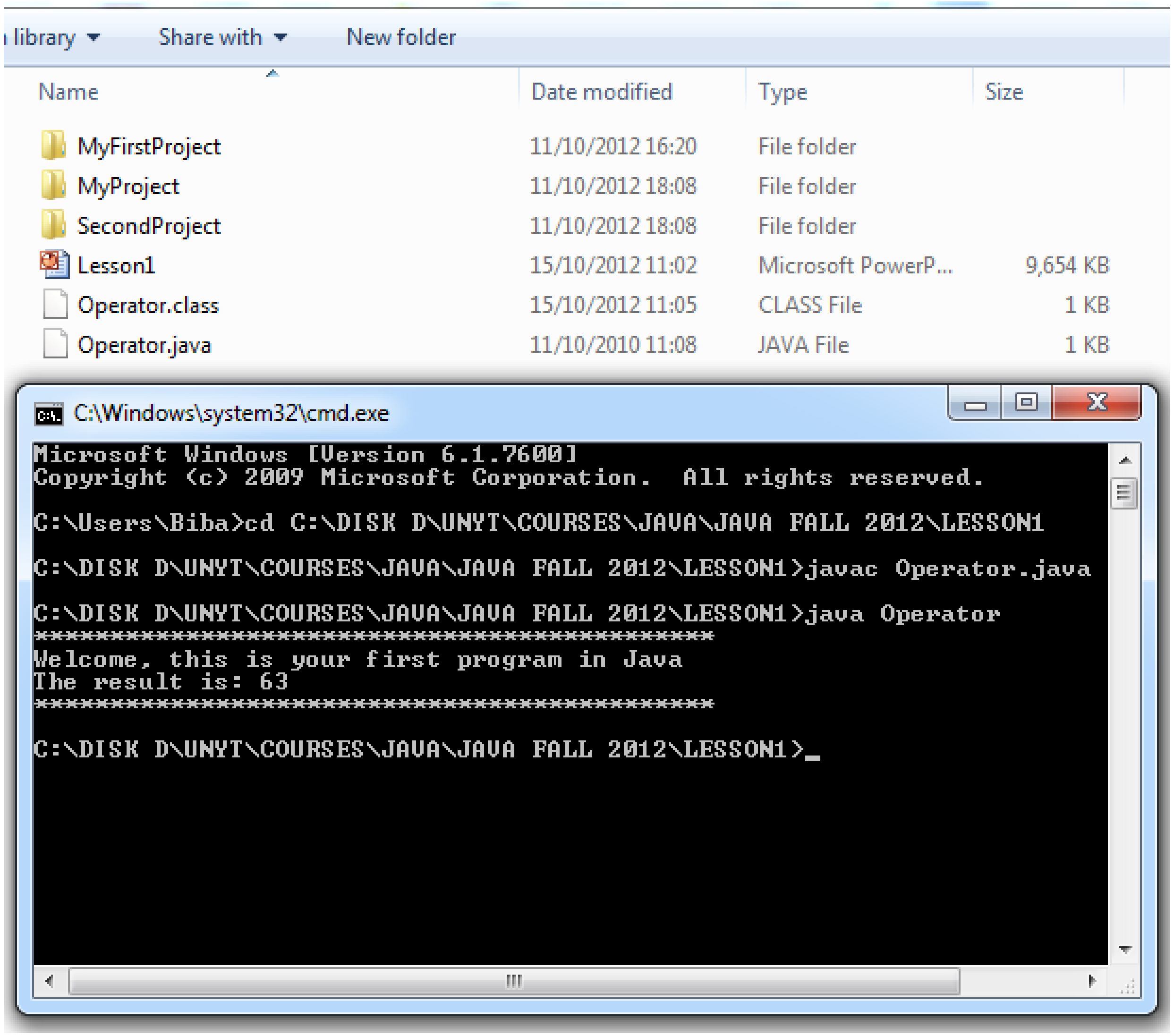


Name	Date modified	Type	Size
MyFirstProject	11/10/2012 16:20	File folder	
MyProject	11/10/2012 18:08	File folder	
SecondProject	11/10/2012 18:08	File folder	
Lesson1	15/10/2012 11:02	Microsoft PowerP...	9,654 KB
Operator.class	15/10/2012 11:05	CLASS File	1 KB
Operator.java	11/10/2010 11:08	JAVA File	1 KB

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

C:\Users\Biba>cd C:\DISK D\UNYT\COURSES\JAVA\JAVA FALL 2012\LESSON1

C:\DISK D\UNYT\COURSES\JAVA\JAVA FALL 2012\LESSON1>javac Operator.java

C:\DISK D\UNYT\COURSES\JAVA\JAVA FALL 2012\LESSON1>
```

.class file generated

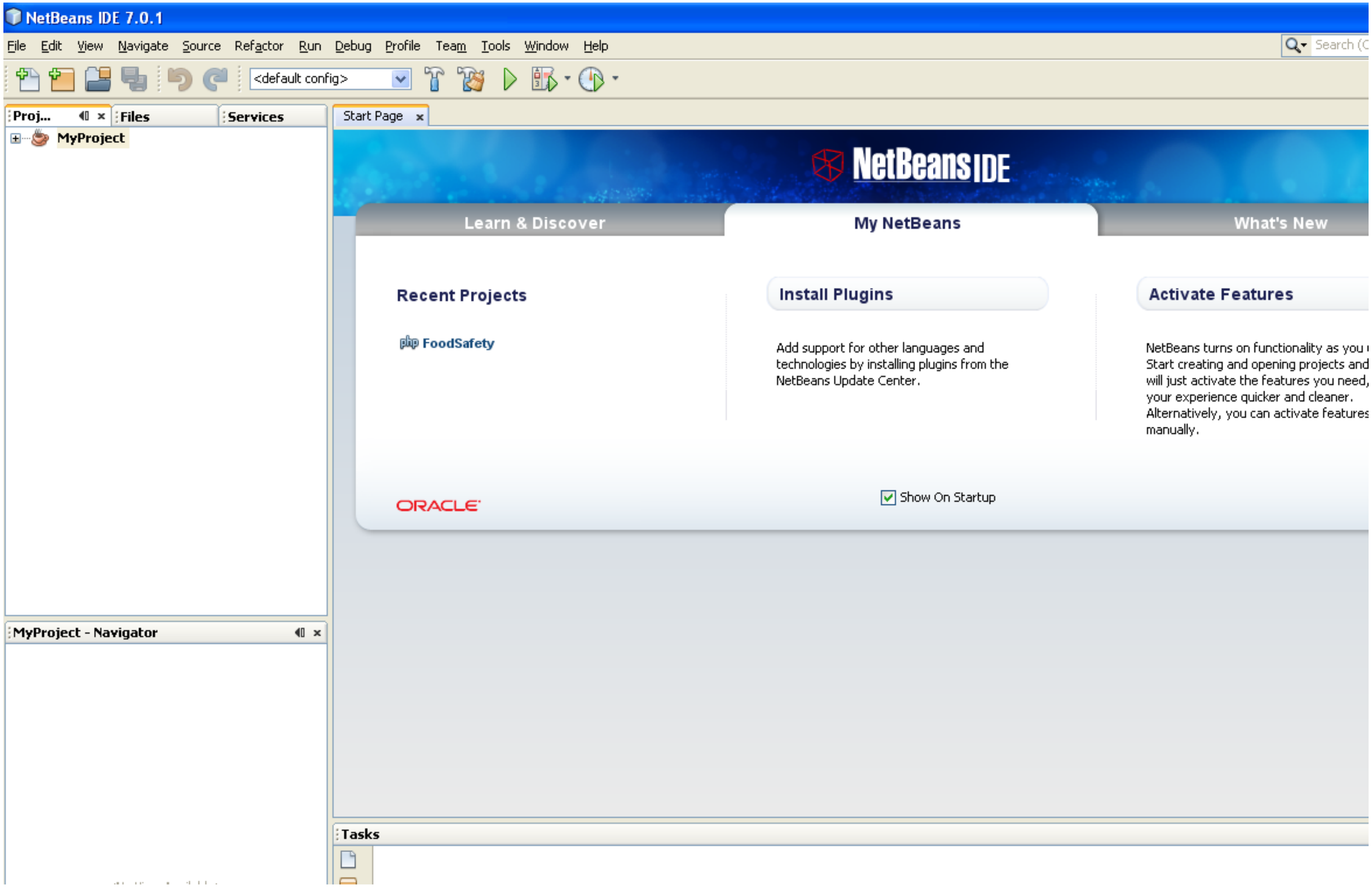
Run the program with the java command



Name	Date modified	Type	Size
MyFirstProject	11/10/2012 16:20	File folder	
MyProject	11/10/2012 18:08	File folder	
SecondProject	11/10/2012 18:08	File folder	
Lesson1	15/10/2012 11:02	Microsoft PowerP...	9,654 KB
Operator.class	15/10/2012 11:05	CLASS File	1 KB
Operator.java	11/10/2010 11:08	JAVA File	1 KB

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

C:\Users\Biba>cd C:\DISK D\UNYT\COURSES\JAVA\JAVA FALL 2012\LESSON1

C:\DISK D\UNYT\COURSES\JAVA\JAVA FALL 2012\LESSON1>javac Operator.java

C:\DISK D\UNYT\COURSES\JAVA\JAVA FALL 2012\LESSON1>java Operator
**********************************************
Welcome, this is your first program in Java
The result is: 63
**********************************************

C:\DISK D\UNYT\COURSES\JAVA\JAVA FALL 2012\LESSON1>_
```

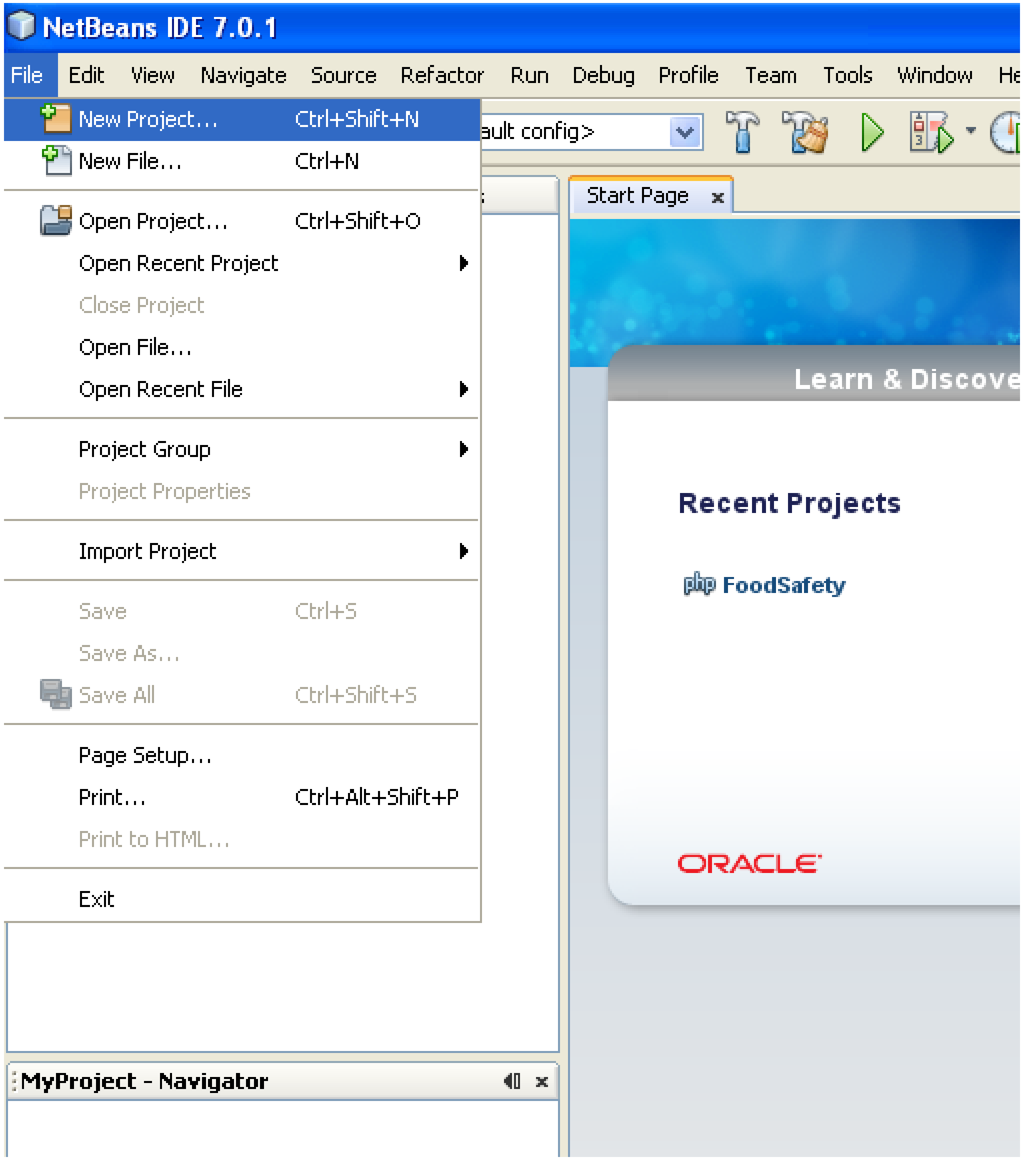
Compiling using Netbeans



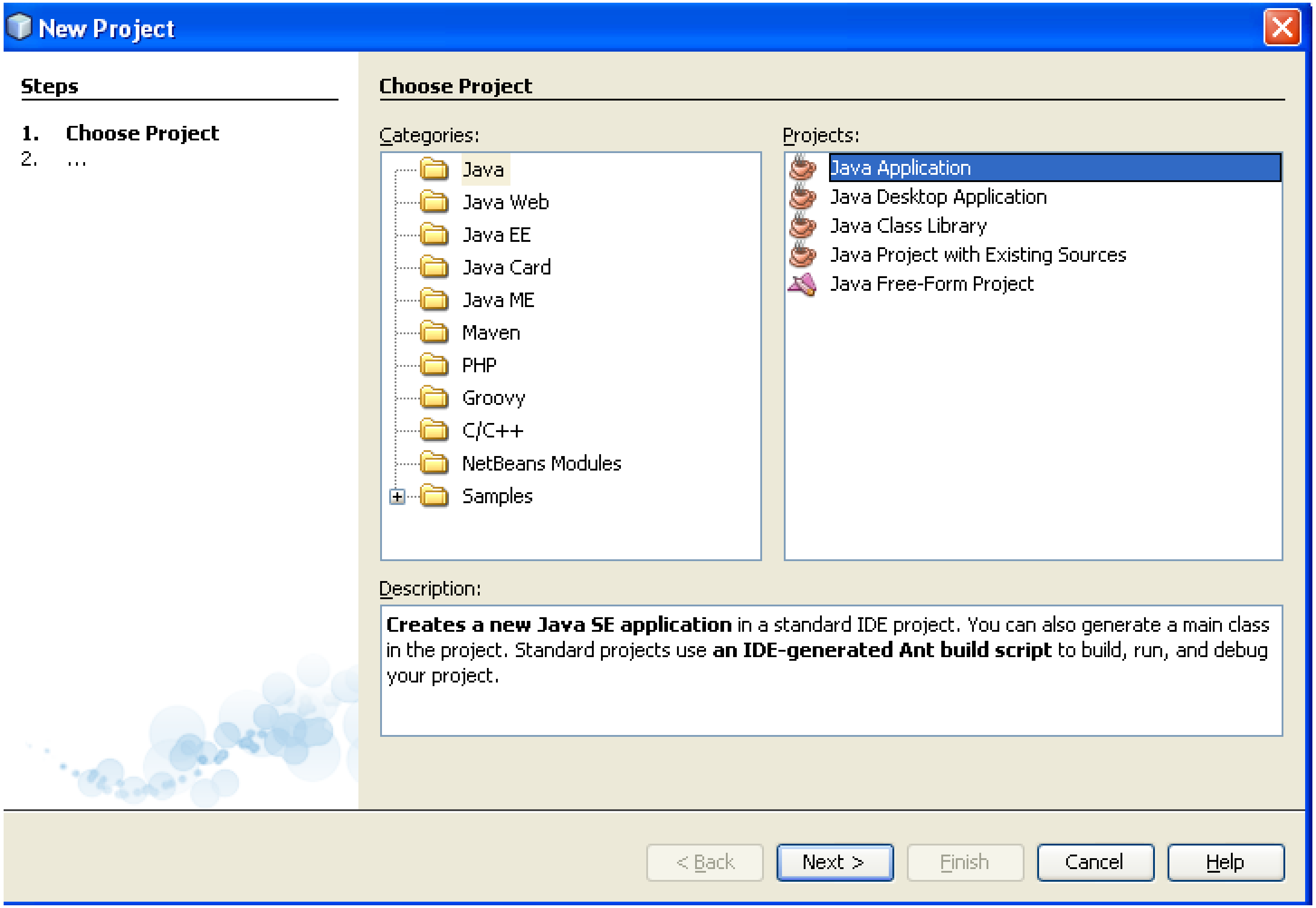
Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

NetBeans turns on functionality as you u
Start creating and opening projects and
will just activate the features you need,
your experience quicker and cleaner.
Alternatively, you can activate features
manually.

New Project


NetBeans IDE 7.0.1
File
Edit
View
Navigate
Source
Refactor
Run
Debug
Profile
Team
Tools
Window
He
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project
Close Project
Open File...
Open Recent File
Project Group
Project Properties
Import Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Page Setup...
Print... Ctrl+Alt+Shift+P
Print to HTML...
Exit
ault config>
Start Page
Learn & Discove
Recent Projects
FoodSafety
ORACLE
MyProject - Navigator

Type of Project



New Project

Steps

1. **Choose Project**
2. ...

Choose Project

Categories:

- Java
- Java Web
- Java EE
- Java Card
- Java ME
- Maven
- PHP
- Groovy
- C/C++
- NetBeans Modules
- Samples

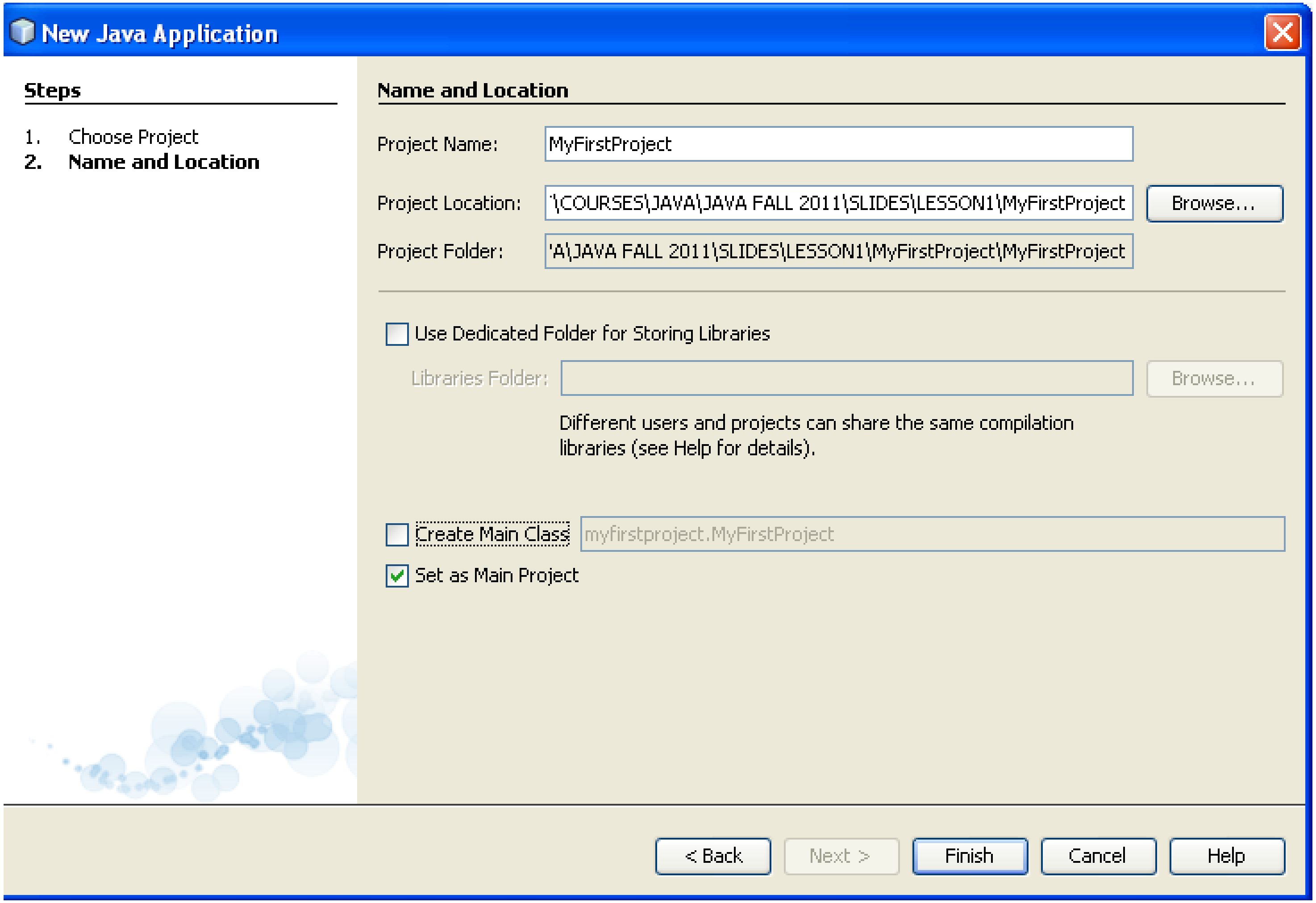
Projects:

- Java Application
- Java Desktop 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 build, run, and debug your project.

Project Name and Location



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: MyFirstProject

Project Location: '\COURSES\JAVA\JAVA FALL 2011\SLIDES\LESSON1\MyFirstProject

Project Folder: 'A\JAVA FALL 2011\SLIDES\LESSON1\MyFirstProject\MyFirstProject

□ Use Dedicated Folder for Storing Libraries

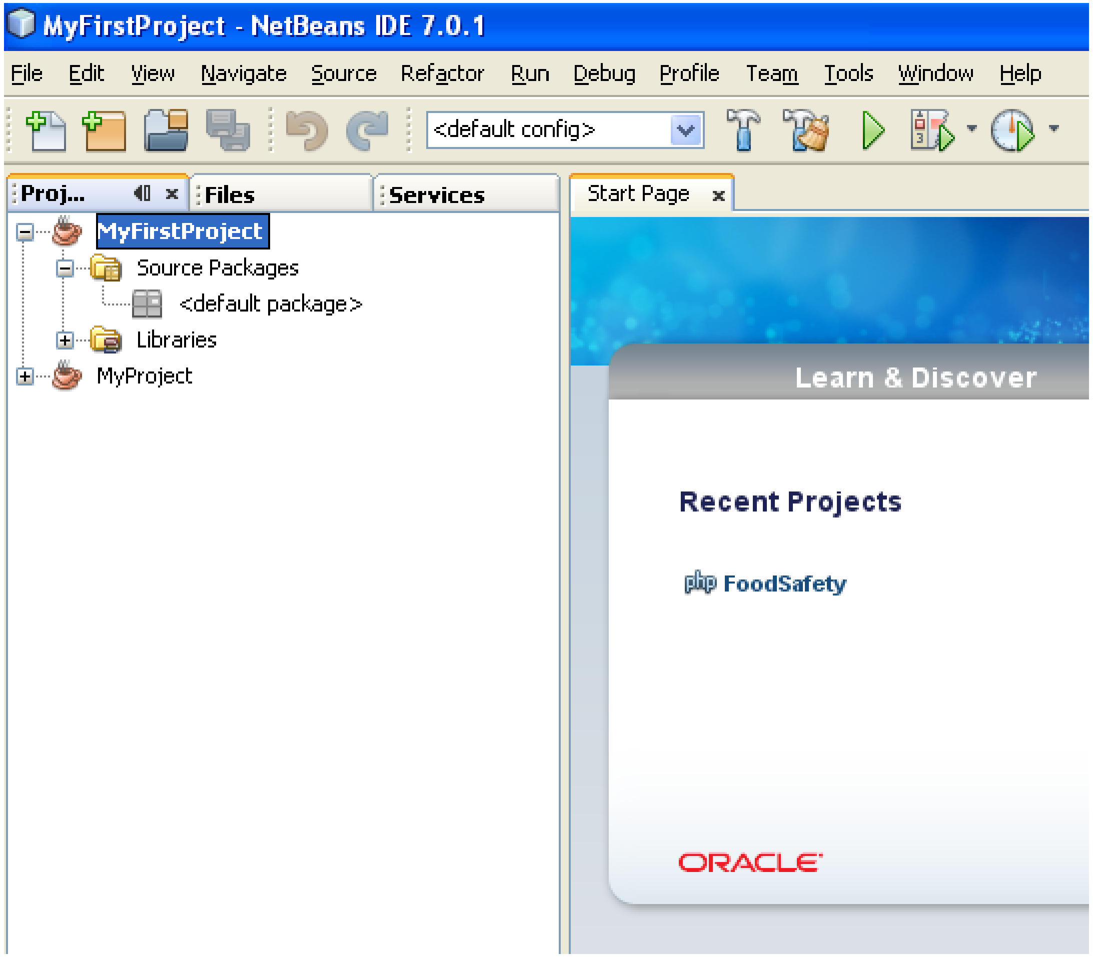
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

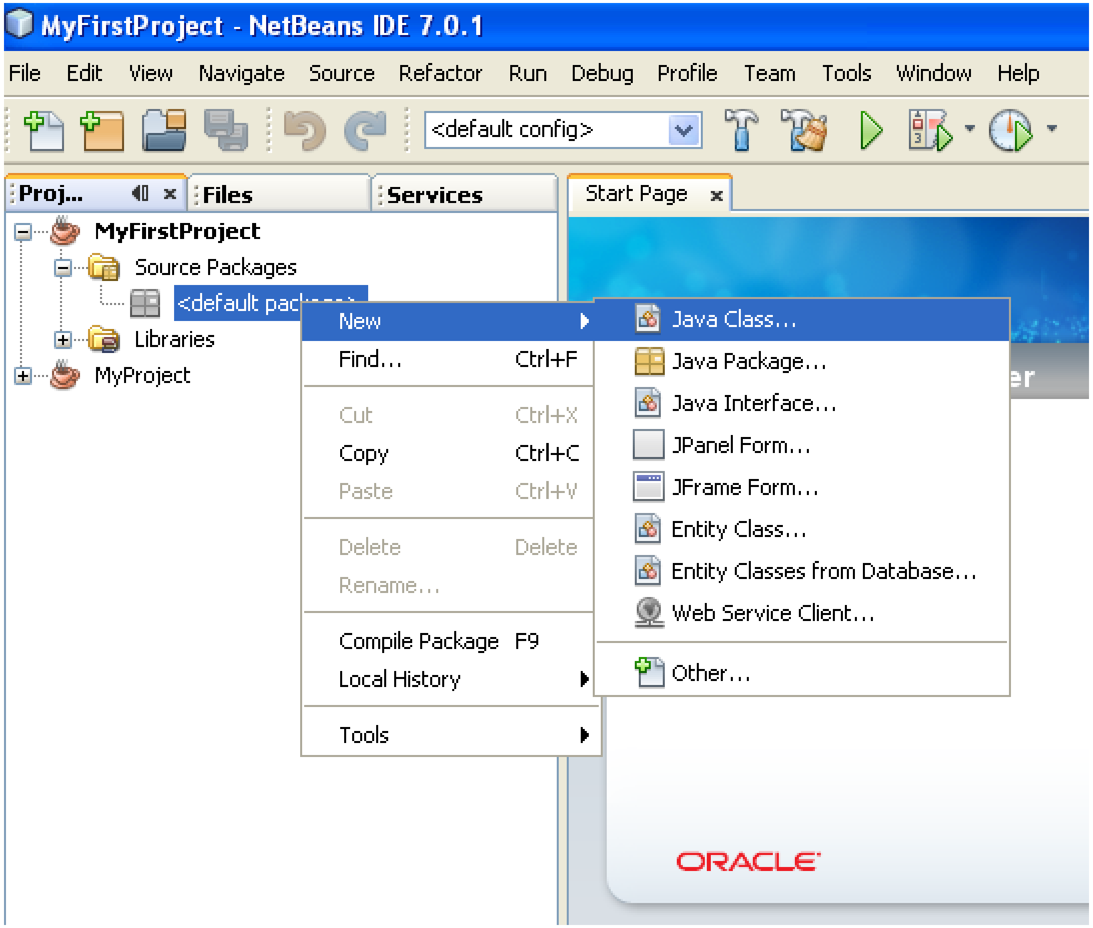
□ Create Main Class myfirstproject.MyFirstProject

☑ Set as Main Project

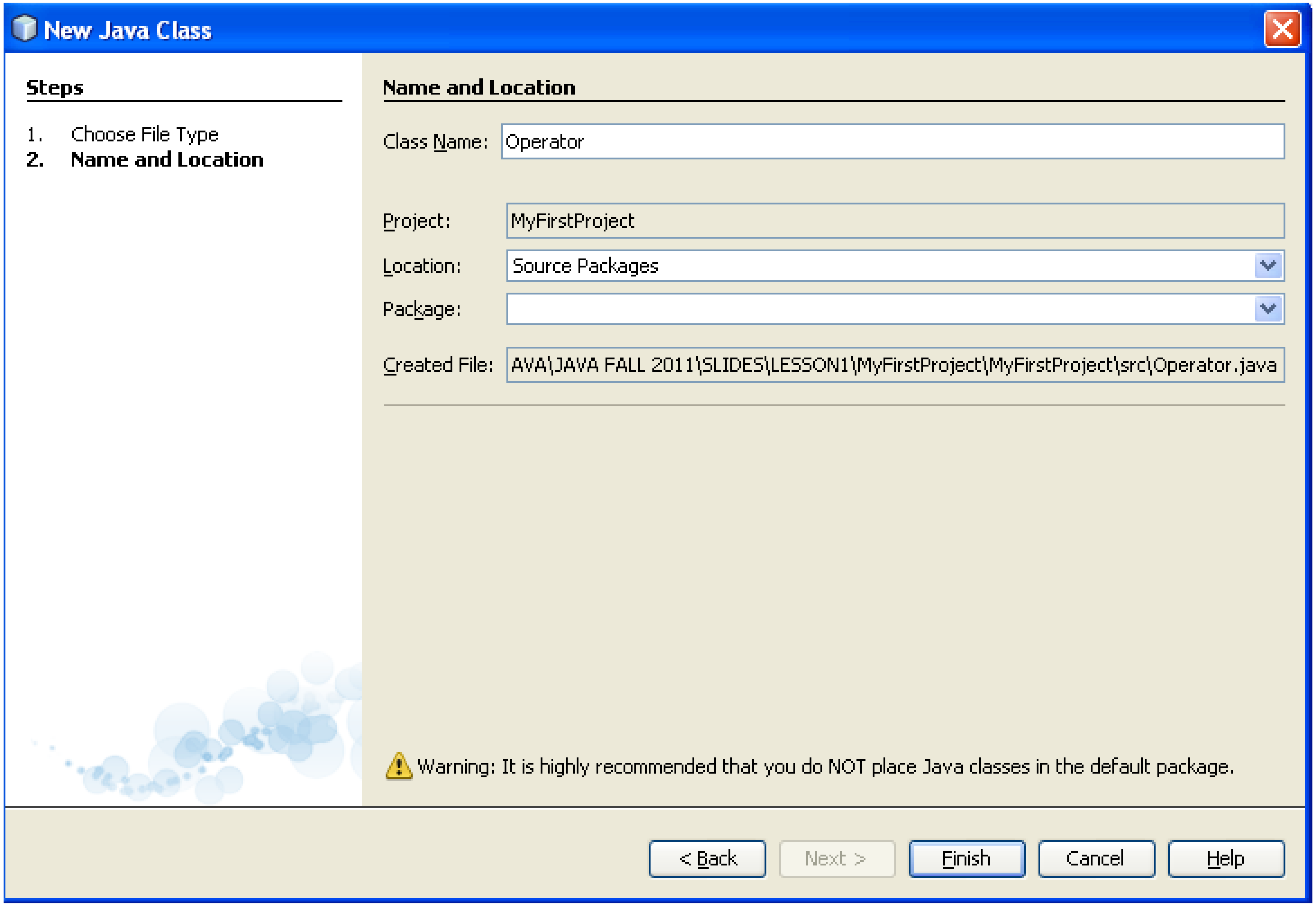
Project created



Create a new class



Create a new class



Class Name: Operator

Project: MyFirstProject

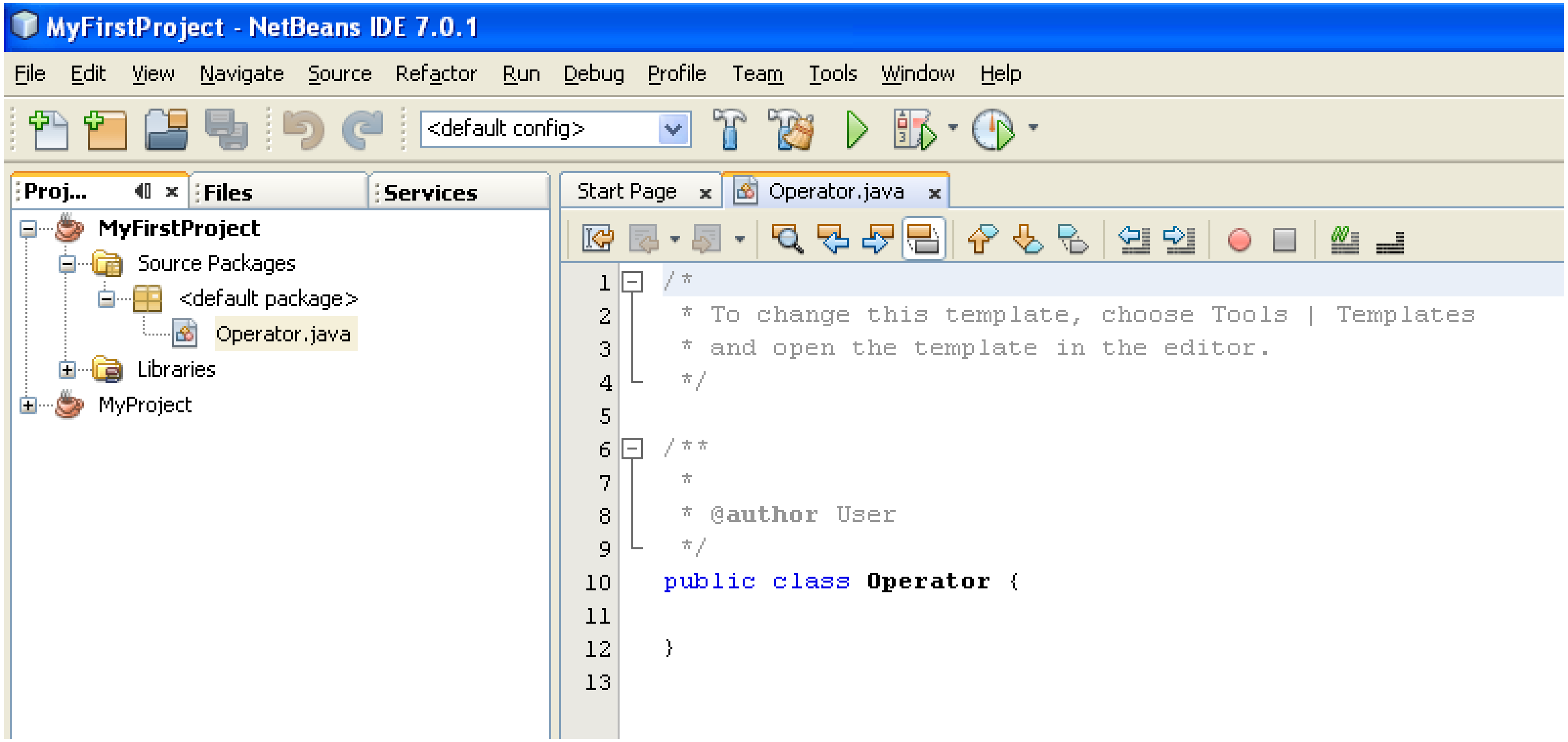
Location: Source Packages

Package:

Created File: AVA\JAVA FALL 2011\SLIDES\LESSON1\MyFirstProject\MyFirstProject\src\Operator.java

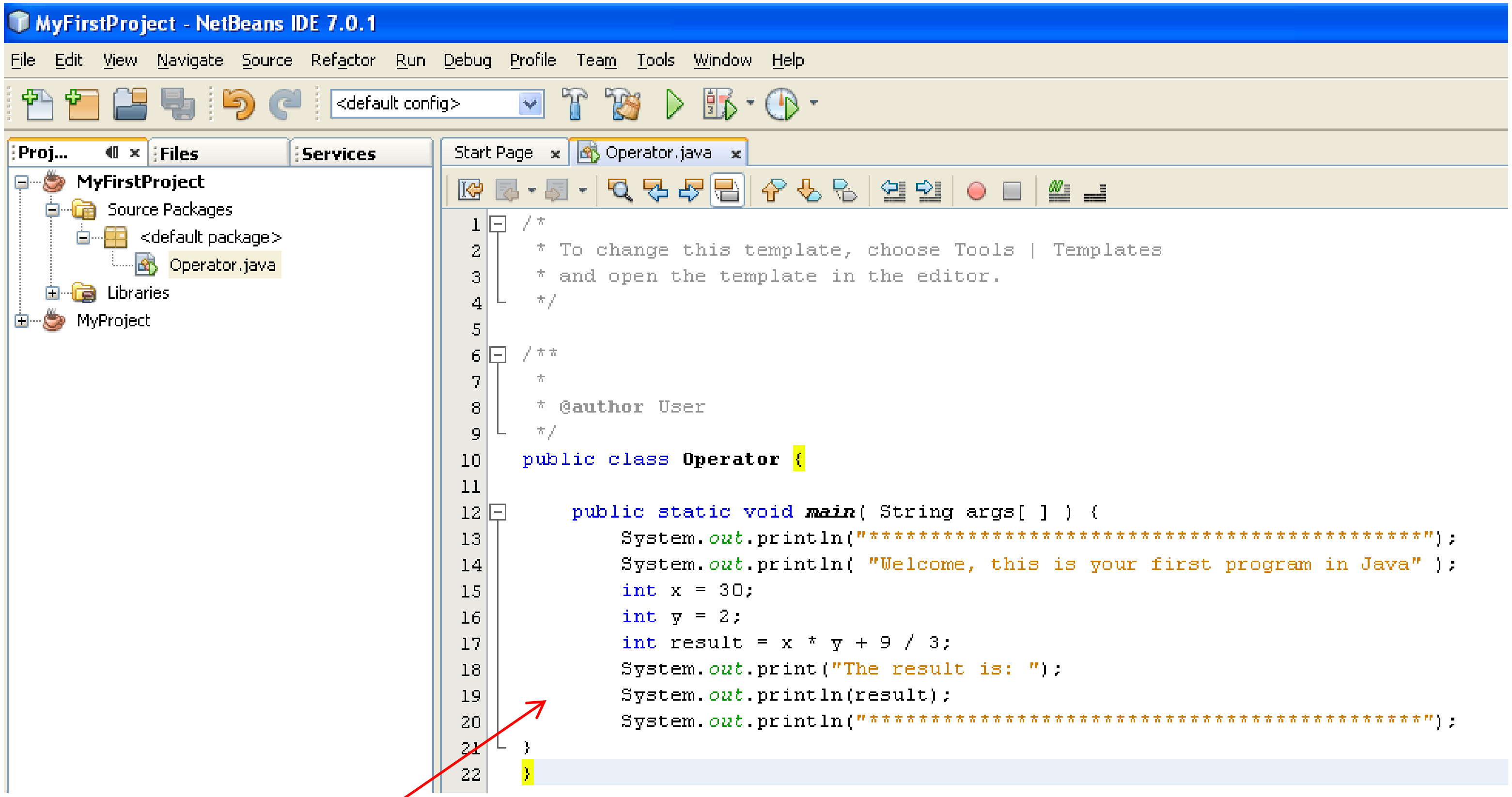
Warning: It is highly recommended that you do NOT place Java classes in the default package.

Write the code



```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author User
 */
public class Operator {

}

```

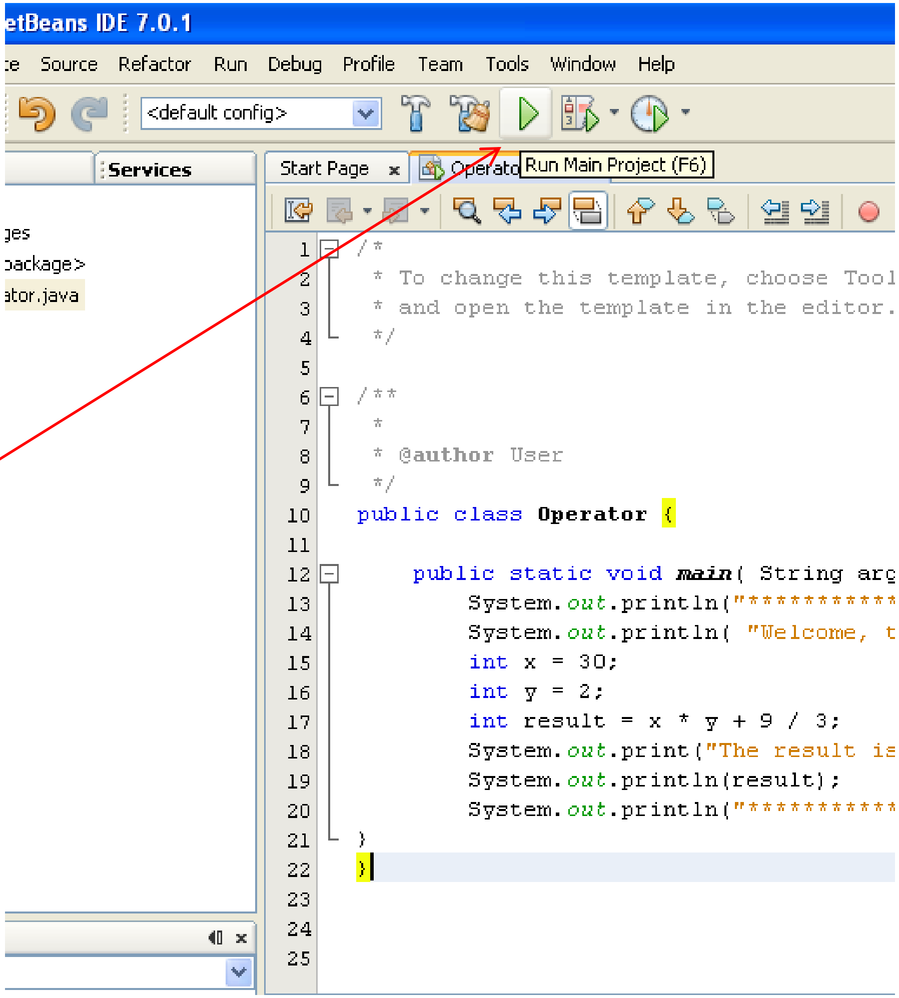
Write the code: start with main



```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author User
 */
public class Operator {

    public static void main( String args[ ] ) {
        System.out.println("*********************************************");
        System.out.println( "Welcome, this is your first program in Java" );
        int x = 30;
        int y = 2;
        int result = x * y + 9 / 3;
        System.out.print("The result is: ");
        System.out.println(result);
        System.out.println("*********************************************");
}
}
```

Write the code

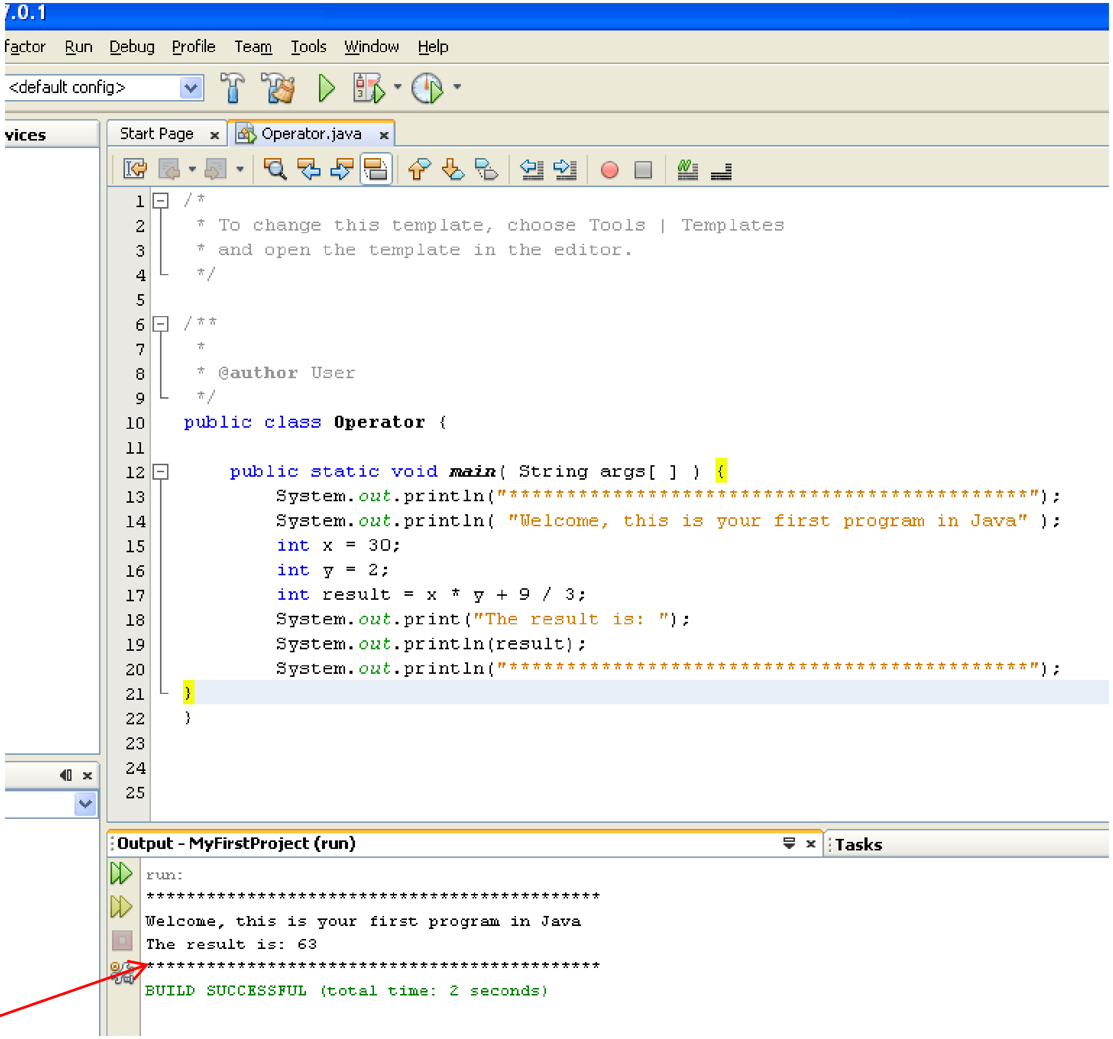
Run the program



```
/*
 * To change this template, choose Tool
 * and open the template in the editor.
 */

/**
 *
 * @author User
 */
public class Operator {

    public static void main( String arg
        System.out.println("***********
        System.out.println( "Welcome, t
        int x = 30;
        int y = 2;
        int result = x * y + 9 / 3;
        System.out.print("The result is
        System.out.println(result);
        System.out.println("***********
}
}
```

Run the program

Console output



```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author User
 */
public class Operator {

    public static void main( String args[ ] ) {
        System.out.println("**********************************************");
        System.out.println( "Welcome, this is your first program in Java" );
        int x = 30;
        int y = 2;
        int result = x * y + 9 / 3;
        System.out.print("The result is: ");
        System.out.println(result);
        System.out.println("**********************************************");
}
}
```

```
run:
**********************************************
Welcome, this is your first program in Java
The result is: 63
**********************************************
BUILD SUCCESSFUL (total time: 2 seconds)
```

Program
output

Exercise 2

```
public class CircleAreaCalculator{

public static void main( String args[ ] ) {
    System.out.println("*********************************************");
    System.out.println( "Welcome, this is your second program in Java" );
    int radius = 10;
    double area = Math.PI * radius * radius;
    System.out.print("The area of the circle is: ");
    System.out.println(area);
    System.out.println("*********************************************");
}
}
```

Exercise 3

- Simple example with object-oriented with Automobiles.
- Auto has Engine, model, color, speed
- Engine has volume and rotations.
- Implement speedup method.

Readings

- Java™ How to Program, 9/e
  - Chapter 1

End of class