Department of Computer Science
Course Code: CSC103
Title: Object-Oriented Programming
Spring 2025

Lab 01

Objectives:
Introduce Java programming, guide system setup for Java development, provide an overview of NetBeans IDE, and cover the basics of Object-Oriented Design and Programming in Java.

Student Information

Student Name	
Student ID	
Date	

Assessment

Marks Obtained	
Remarks	
Signature	

Lab-01
Setting up the Machine for Java

Objective

Introduce Java programming, guide system setup for Java development, provide an overview of NetBeans IDE, and cover the basics of Object-Oriented Design and Programming in Java.

Apparatus:

Hardware requirements:
Dual core CPU based on x64 architecture
Minimum of 1 GB RAM
800 MB of disk space
Software requirements:
Windows 7 SP1 (64 bit)
Java JDK 8 (64 bit JVM)
NetBeans IDE (version 8.1 or above)

Background

1.1 Programming Languages

By programming language, we mean a set of vocabulary, grammar and libraries that construct a language we can use to write computer programs. A programming language may consist of the following:

Vocabulary: set of meaningful keywords that have specific meaning in the programming language (also called reserved words).

Syntax (grammar): set of rules that constraint the way in which programs are written. A program that does not follow these rules has one or more syntax errors.

Software Development Kit (SDK) that contains the following:

Libraries: also known as Application Programming Interface (API), these files are previously written classes and methods that contain some common functionality.

Compiler: the program that translates files written in Java language (human language) into binary files (machine language) in order for the computer to be able to execute them.

Interpreter: some programming languages does not compile the source code file into directly executable form (native code), but they instead compile it into partially compiled file that could be executed by a program called interpreter.

Procedure:

1.2 Java Programming Language

Java programming language is an interpreted programming language, that is, when the source code is compiled into binary file, it needs an interpreter called Java Virtual Machine (JVM) to run it. Java compiler is called javac.exe, and interpreter is called java.exe. Figure 2.1 shows a Java technology overview.

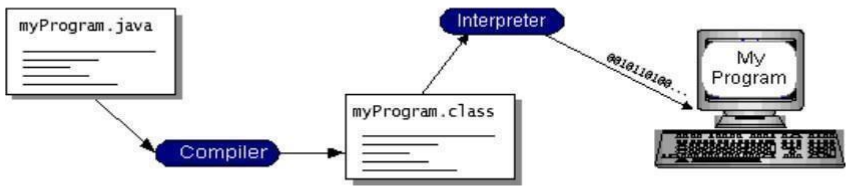


Figure 1.1: Java technology

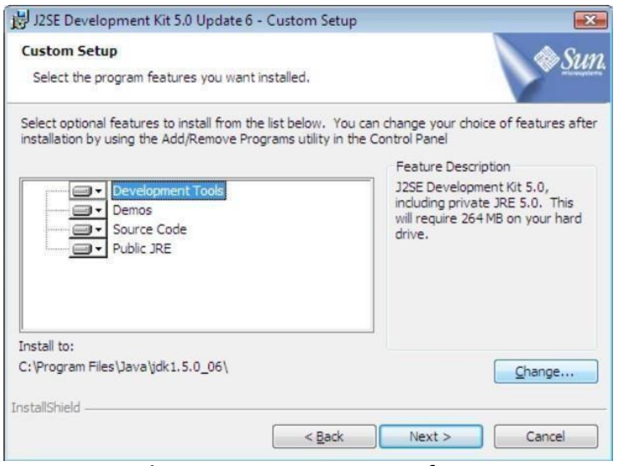
1.3 Setting-up Your Machine for Java Development

To be able to write, compile and execute Java applications, you need a set of tools called Java Software Development Kit (JSDK). This SDK could be obtained from Java official site at. After downloading the installation file you are ready to start the installation process. Follow the following procedure to prepare your machine.

Step 1: Double click on executable java setup file icon, the setup wizard appears as in

Figure 1.2: First step of JDK setup wizard

Select I accept the terms in the license agreement and click on Next. Custom setup dialog appears as in figure 1.3.


J2SE Development Kit 5.0 Update 6 - Custom Setup

Custom Setup

Select the program features you want installed.

Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

Feature Description

J2SE Development Kit 5.0, including private JRE 5.0. This will require 264 MB on your hard drive.

Install to:

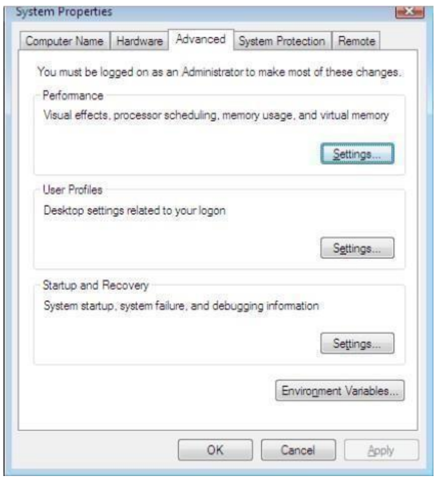
C:\Program Files\Java\jdk1.5.0_06\

Figure 1.2: First step of JDK

Figure 1.3: Setup options

It is advisable that you change the installation path by clicking on Change button andselecting the path "C:\jdk"as installation directory. After you have done changing the path, click on Next to start the installation and wait until it‘s done. When installation is complete, click on Finish to terminate installation wizard.

Step 2: You need now to define some environment variables to be able to use the tools you've just installed. To define an environment variable, right-click on My Computer icon on the desktop or start menu, then select Properties from the pop-up menu. The System Properties dialog appears from the top of the dialog box select Advanced tab as shown in figure 1.4. From Advanced tab, click on Environment Variables button, Environment Variables dialog appears as shown in figure 1.5.



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

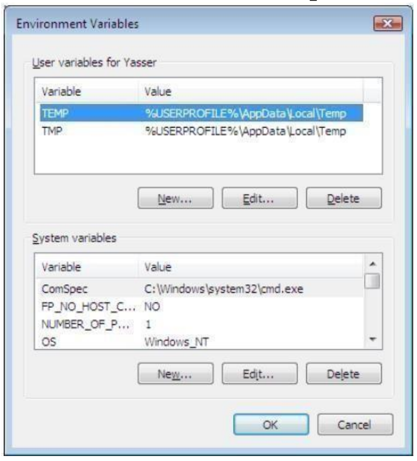
Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your logon

System startup, system failure, and debugging information

Figure 1.3: Setup options

Figure 1.4: Windows advanced settings
To add new environment variable, click on New button in User Variables field, note that you may not modify system variables unless you have administrator privileges. Once you click on New button, variable addition dialog appears as in figure 1.6.
In Variable name field enter the name of your variable, in Variable value field enter the value of the variable then click on OK. You need to add two variables to use with JDSK:
● PATH: points to bin directory in JSDK home directory (e. g. c:\jdk\bin)
● CLASSPATH: points to the folder in which you will create you Java programs (e. g.c:\Documentsand Settings\Student\Desktop\java).
Your computer is ready now to be used for Java development.



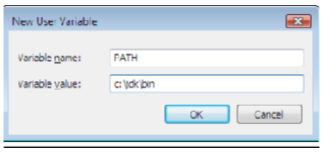
User variables for Yasser

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

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	1
OS	Windows_NT

Figure 1.4: Windows advanced settings
Figure 1.5: Environment Variables



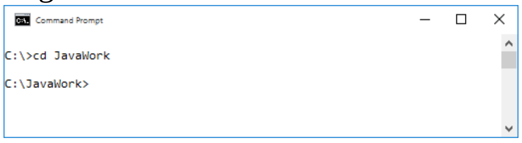
Variable name: PATH
Variable value: c:\jdk\bin

Figure 1.5: Environment Variables

Figure 1.6: Adding new environment variable dialog box

1.4 Writing, Compiling and Running Your First Java Application

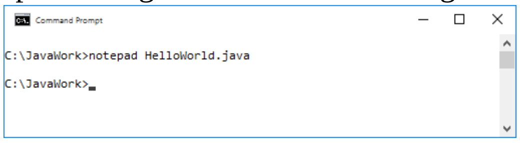
To write a Java program, open the command prompt by selecting run from start menu and entering the command cmd in the dialog box, the command prompt console appears, show in figure 1.7. Change to the directory you have selected as Java directory (for examplec:\JavaWork) by using cd command.



```
C:\>cd JavaWork

C:\JavaWork>
```

Figure 1.6: Adding new environment variable dialog box

To write a new Java program using the notepad, enter the command notepad filename. For instance, we will write a class named HelloWorld, so we will use the command notepad HelloWorld.java (Figure 1.5). Note that we use a .java extension for all Java source files, and we always use upper case characters in the beginning of a class name. After starting the notepad, it will ask you whether you wish to create a new file with the name specified, confirm file creation. It is important to keep in mind that file name and class name must be the same, otherwise the compiler will generate an error message and the file will not compile.



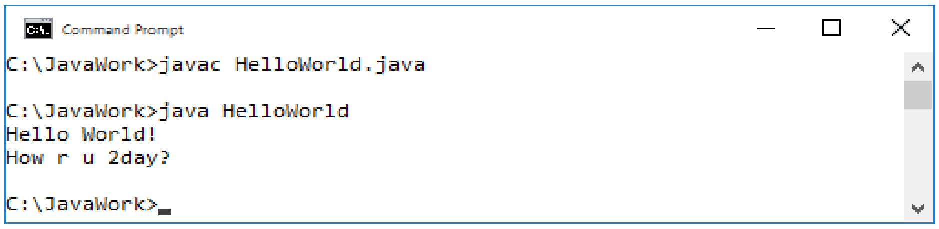
```
C:\JavaWork>notepad HelloWorld.java

C:\JavaWork>_
```

Figure 1.8: notepad command with parameters

In the notepad, write the following code (always keep in mind that Java is case-sensitive language, for example if you typed Class with capital _C‘ instead of class, the compiler will report a syntax error)

```
//HelloWorld.java
/* This is my first java program.
*/
class HelloWorld{
public static void main(String[] args){
System.out.println("Hello   World!");
System.out.println("How r u 2day?");
} //end of main method
} //end of class
```

When you are done with code writing, save the file and exit notepad. Your source code file is now ready for compilation, to compile a Java source code file, use the command javac filename. In our example, javac HelloWorld.java. If everything is alright, the compiler will terminate without any messages and go back to command prompt as shown in figure 1.6. After compilation, if you run dir command, you will notice that the compiler generated a newfile called HelloWorld.class, this is the binary file JVM will execute. To run you application,use the command java Class Name , for our program, we will use java HelloWorld, if everything is alright, you shall see the program output on the screen.
Notice that some lines in the program begin with //, these are called comments, compiler does not read comments, so whatever your write in them it will not affect the functionality of your program, the notations /* and */ declare the beginning and ending of a block of comments, sowe call them multi-line comments.



```
C:\JavaWork>javac HelloWorld.java

C:\JavaWork>java HelloWorld
Hello World!
How r u 2day?

C:\JavaWork>_
```

Figure 1.9: successful compilation

```
System.out.println("Hello World");
```

System is a name of a class, inside System class, there is a public and static member (object)ofclass PrintWriter, the name of this object is out. Inside PrintWriterclass, there is a public, non-static method called println that has several overloads.

1.5 IDE Software

IDE stands for Integrated Development Environment, IDE is a software that helps us to write, compile, run and debug programs. IDE programs vary and each one of them usually targeted to certain programming language, the following table shows well-known IDE programs:

Programming Languages	IDE
Java	• NetBeans • Forte for Java • Borland JBuilder
C/C++	• Microsoft Visual Studio • Borland C++
PHP/JSP/ASP	• Macromedia Dreamweaver
ASP.NET	• Microsoft Visual Studio .NET • Sharp Develop

Tabel 1.1: Programming languages and respective IDE

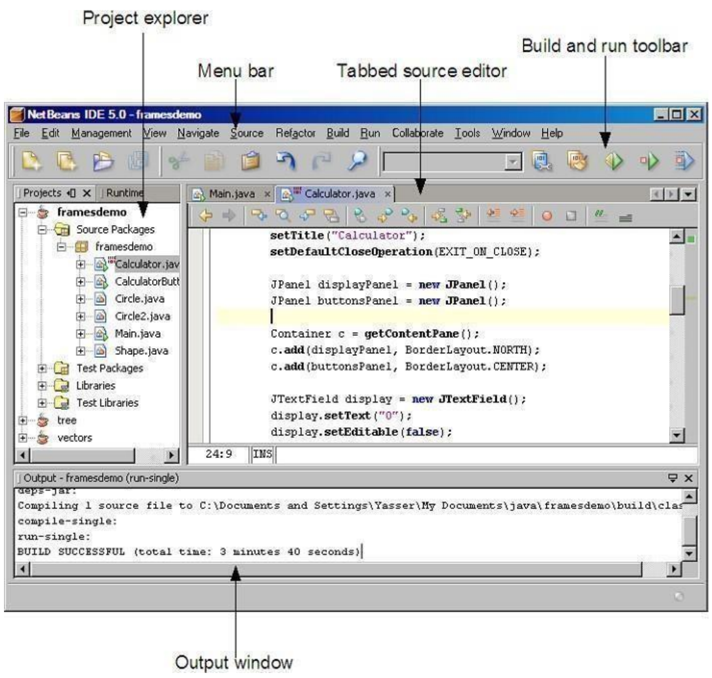
We can notice from table above that we have various brands of IDEs, since we are interested in Java programming language in this course, we will focus on its IDEs. NetBeans is the best offered IDE for us as Java learners, because it is free compared to JBuilder, more advanced and stable compared to Forte. So lets start having a look at NetBeans.

1.6 NetBeans IDE

Figure shows the main screen of NetBeans IDE (version 8.1). You can download NetBeans from http://java.sun.com, once you have latest JDK version installed, NetBeans installation is a straight-forward process.

To use NetBeans for writing and running your Java programs, follow these steps:

1. Create a new project by going to File > New Project, or by clicking on New Project button on the toolbar; all programs written using NetBeans must be included in projects to be able to be compiled and run. Project creation is a simple process consists of two steps. First step is defining project category and type. In our case, we leave the default settings, that is, General category and Java Application type. Second steps specifying project name and main class.



```
        setTitle("Calculator");
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        JPanel displayPanel = new JPanel();
        JPanel buttonsPanel = new JPanel();

        Container c = getContentPane();
        c.add(displayPanel, BorderLayout.NORTH);
        c.add(buttonsPanel, BorderLayout.CENTER);

        JTextField display = new JTextField();
        display.setText("0");
        display.setEditable(false);
```

```
deps-jar:
Compiling 1 source file to C:\Documents and Settings\Yasser\My Documents\java\framesdemo\build\clas
compile-single:
run-single:
BUILD SUCCESSFUL (total time: 3 minutes 40 seconds)
```

Figure 1.10: Screenshot of Java IDE

2. After creating a project, you are ready to start coding, on the project explorer you will see a tree that has the project name you specified as root. Project is split into four folders: Source packages, Test packages, Libraries and Test libraries. Source packages is the folder containing the source code files (.java files) and hence it is the folder we are interested in. Source code is usually split into packages, so we can group classes of related functionality together. Once you create a project, a default package that holds project name is created.

3. In the code editor, you will see the created Main class, with its main method already coded, so you need just to start programming.

4. In the main method, try to write some statements you've learned from session 1, after you are done, you may compile and run the program by selecting Run > Run main project, pressing F6 key on the keyboard, or clicking Run button on the toolbar

5. The output window appears at the bottom of the IDE and shows the outputs of compilation and running processes.

Introducing Object Oriented

Everyone knows what an object is—a tangible thing that we can sense, feel, and manipulate. The earliest objects we interact with are typically baby toys. Wooden blocks, plastic shapes, and oversized puzzle pieces are common first objects. Babies learn quickly that certain objects do certain things: bells ring, buttons press, and levers pull. The definition of an object in software development is not terribly different. Software objects are not typically tangible things that you can pick up, sense, or feel, but they are models of something that can do certain things and have certain things done to them. Formally, an object is a collection of data and associated behaviors. So, knowing what an object is, what does it mean to be object-oriented? Oriented simply means directed toward. So object-oriented means functionally directed towards modeling objects. This is one of the many techniques used for modeling complex systems by describing a collection of interacting objects via their data and behavior . If you've read any hype, you've probably come across the terms object-oriented analysis, object oriented design, object-oriented analysis and design, and object oriented programming. These are all highly related concepts under the general object-oriented umbrella.

In fact, analysis, design, and programming are all stages of software development. Calling them object-oriented simply specifies what style of software development is being pursued.
Object-oriented analysis (OOA) is the process of looking at a problem, system, or task (that somebody wants to turn into an application) and identifying the objects and interactionsbetween those objects. The analysis stage is all about what needs to be done. The output of the analysis stage is a set of requirements. If we were to complete the analysis stage in one step, wewould have turned a task, such as, I need a website, into aset of requirements.
For example: Website visitors need to be able to (italic represents actions, bold represents objects):

- review our history
- apply for jobs
- browse, compare, and order products

In some ways, analysis is a misnomer. The baby we discussed earlier doesn't analyze the blocks and puzzle pieces. Rather, it will explore its environment, manipulate shapes, and see where they might fit. A better turn of phrase might be object-oriented exploration. In software development, the initial stages of analysis include interviewing customers, studying their processes, and eliminating possibilities. Object-oriented design (OOD) is the process of converting such requirements into an implementation specification. The designer must name the objects, define the behaviors, and formally specify which objects can activate specific behaviors on other objects. The design stage is all about how things should be done. The output of the design stage is an implementation specification. If we were to complete the design stage in a single step, we would have turned the requirements defined during object-oriented analysisinto a set of classes and interfaces that could be implemented in (ideally) any object-oriented programming language.

Introduction to Object Oriented Programming with Java

Object-oriented programming (OOP) is a way of thinking about and organizing code for maximum reusability. With this type of programming, a program comprises objects that can interact with the user, other objects, or other programs. This makes programs more efficient and easier to understand.

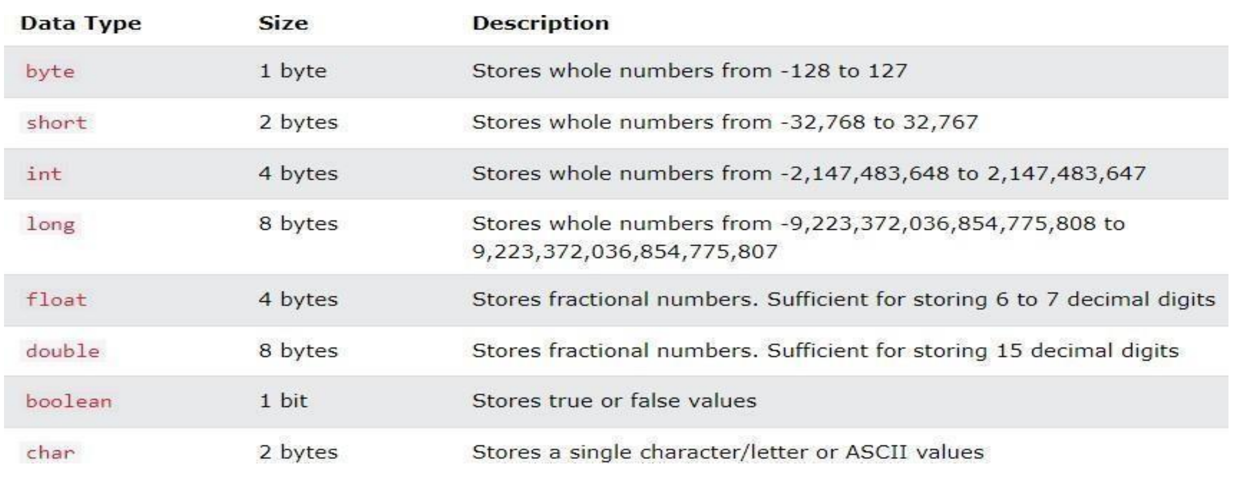
Object-Oriented Programming (OOP) is a programming paradigm that is widely used in Java and many other programming languages. It revolves around the concept of objects, which are instances of classes, and the organization of code into reusable and modular structures. Java is a popular language for OOP, and it provides strong support for OOP principles.

Characteristics of Object Oriented Languages.

The major elements of object-oriented languages in general are;

- Objects
- Classes
- Inheritance.
- Reusability
- Abstraction
- Encapsulation
- Polymorphism

Data Types in Java Basic Data Types:



Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

A Basic Java Program.

```
public class HelloWorld {
public static void main(String[] args) {
    System.out.println("Hello, World!");
}
 }
```

Java User Input

The Scanner class is used to get user input, and it is found in the java.util package.To use the Scanner class, create an object of the class and use any of the available methods found in the Scanner class documentation.

```
import java.util.Scanner; // Import the Scanner class

public class UserInputExample {
    public static void main(String[] args) {
        // Create a Scanner object to read user input
        Scanner scanner = new Scanner(System.in);

        // Prompt the user for input
        System.out.println("Enter your name: ");

        // Read the user's input as a string
        String userName = scanner.nextLine();

        // Display a greeting using the user's input
        System.out.println("Hello, " + userName + "!");

        // Close the scanner
        scanner.close();
    }
}
```

Lab Exercises

1. Write a Java program that prompts the user to enter their birth year and then calculates and displays their age.
2. Write a program which inputs your name and prints its length on the display screen. For input, use Scanner class.
3. Write a program which takes weight in KGs and converts it into pounds. For input, use Scanner class.(1 Kg = 2.2 Pounds)