**Departmentof ComputerScience Course
Code: CSC103
Title:Object-OrientedProgramming
Spring 2025**

Lab01

Objectives:

IntroduceJavaprogramming,guidesystemsetupforJavadevelopment,provideanoverviewof NetBeans IDE, and cover the basics of Object-Oriented Design and Programming in Java.

StudentInformation

StudentName	Abdul Rehman Ali
StudentID	24FA-025-SE
Date	

Assessment

MarksObtained	
Remarks	
Signature	

Lab-01
SettingupationeMachineforJava

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:

Hardwarerequirements:
DualcoreCPUbasedonx64architecture Minimum
of 1 GB RAM
800MBofdisk space

Softwarerequirements:
Windows7 SP1 (64 bit)
JavaJDK8(64bitJVM)
NetBeans IDE(version8.1orabove)

Background

ProgrammingLanguages

By programming language, we mean a set of vocabulary, grammar and libraries that constructalanguagewecanusetowritecomputerprograms.Aprogramminglanguagemay consist of the following:

Vocabulary:setofmeaningfulkeywordsthathavespecificmeaningintheprogramming language (also called reserved words).

Syntax(grammar):setofrulesthatconstraintthewayinwhichprogramsarewritten.A program that does not follow these rules has one or more syntax errors.

SoftwareDevelopment Kit(SDK)thatcontainsthefollowing:

Libraries:alsoknownasApplicationProgrammingInterface(API),thesefilesare previously written classes and methods that contain some common functionality.

Compiler:theprogramthattranslatesfileswritteninJavalanguage(humanlanguage)intobinary 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:

JavaProgrammingLanguage

Javaprogramminglanguageisaninterpretedprogramminglanguage,thatis,whenthesource codeiscompiledintobinaryfile,itneedsaninterpretercalledJavaVirtualMachine(JVM)to run it. Javacompileris called javac.exe, and interpreteris called java.exe. Figure2.1 shows a Java technology overview.

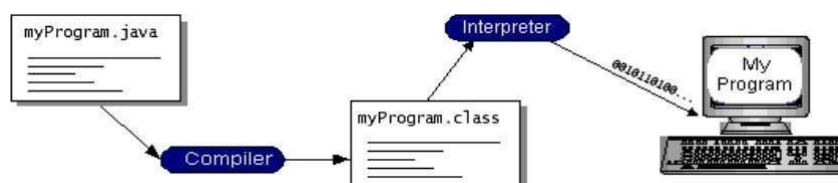**Figure1.1:**Javatechnology

Setting-upYourMachineforJavaDevelopment

Tobeableto write, compileand executeJavaapplications, you need asetoftools called

Java Software Development Kit (JSDK). This SDK could be obtained from Java official site at.Afterdownloadingtheinstallationfileyouarereadytostarttheinstallationprocess.Follow the following procedure to prepare your machine.

Step1:Doubleclickon executablejavasetupfileicon, thesetupwizardappearsas in

Figure1.2:First stepofJDKsetup wizard

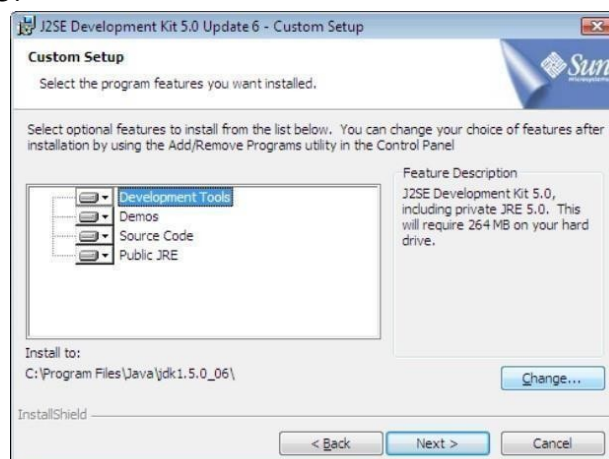Select I accept the terms in the license agreement and click on Next. Custom setup dialog appears as in figure 1.3.

Figure1.2: FirststepofJDK

Figure1.3:Setupoptions

It is advisable that you change the installation path by clicking on Change button andselecting thepath"C:\jdk"asinstallationdirectory.Afteryouhavedonechangingthepath,clickonNext to start the installation and wait until it's done. When installation is complete, click on Finish to terminate installation wizard.

Step2: You need now to define some environment variablesto be able touse the tools you've just installed. To define an environment variable, right-click on My Computer icon on thedesktoporstartmenu,thenselectPropertiesfromthepop-upmenu.TheSystemProperties dialog appears from the top of the dialog box select Advanced tab as shown in figure 1.4.

FromAdvancedtab,click onEnvironment Variablesbutton,EnvironmentVariablesdialog appears as shown in figure 1.5.
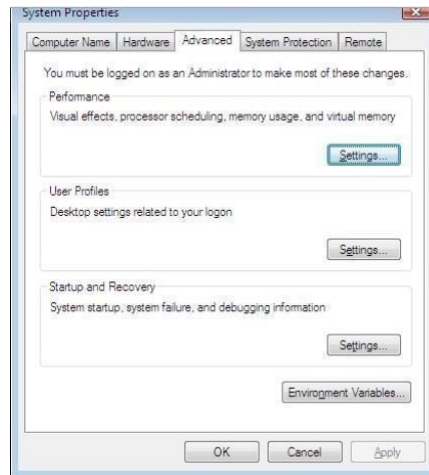
Figure1.3:Setup options

Figure1.4:Windowsadvancedsettings

To add new environment variable, click on New button in User Variables field, note that you may not modify systemvariables 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 yourvariable, 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:pointstobindirectoryinJSDKhomedirectory(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).
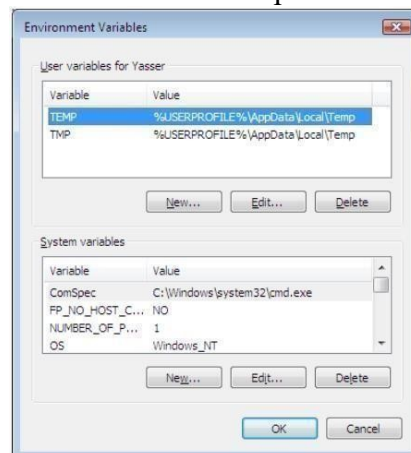
Yourcomputer is readynow tobeused forJavadevelopment.

Figure1.4: WindowsadvancedsettingsFigur
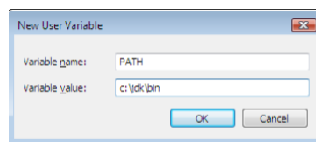e 1.5: Environment Variables

Figure1.5:EnvironmentVariables Figure

1.6: Adding new environment variable dialog box

Writing,Compiling andRunningYourFirstJavaApplication

To write a Java program, open the command prompt by selecting run from start menu and enteringthecommand cmdinthedialogbox,thecommandpromptconsoleappears,showin

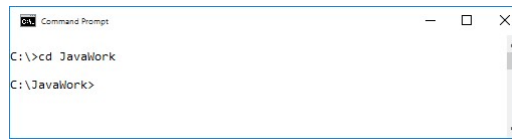figure1.7.Changetothedirectoryyouhaveselectedas Javadirectory(for examplec:\JavaWork) by using cd command.

Figure1.6: Addingnewenvironment variabledialogbox

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(Figure1.5).Notethatweusea.javaextensionforallJavasourcefiles,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 thesame,otherwisethecompilerwillgeneratean errormessageandthefilewillnotcompile.

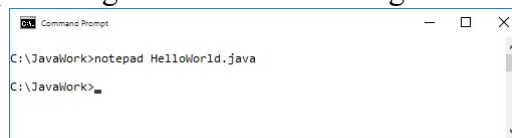Figure1.8:notepadcommandwith parameters

In the notepad, write the following code (always keep in mind that Java is case-sensitive language,forexampleifyoutypedClasswithcapital_C'insteadofclass,thecompilerwill report a syntax error)

```
//HelloWorld.java
/*This ismyfirstjavaprogram.
*/
classHelloWorld{
publicstaticvoidmain(String[]args){ S
ystem.out.println("Hello    World!");
System.out.println("Howru2day?");
}//end ofmain method
}//end ofclass
```

Whenyouaredonewithcodewriting,savethefileandexitnotepad.Yoursourcecodefileis 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.

Aftercompilation,ifyourundircommand,youwillnoticethatthecompilergenerateda 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,compilerdoes 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.
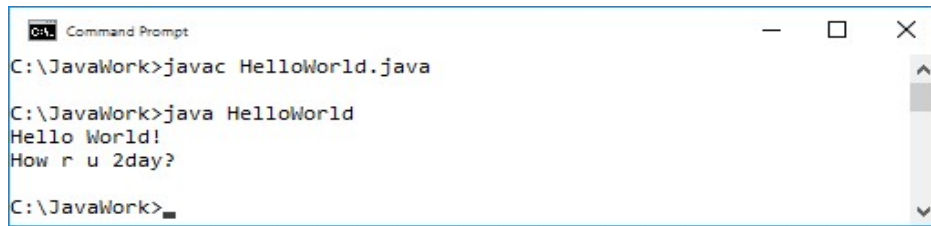
Figure1.9:successfulcompilation

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

System is a name of a class, inside System class, there is a public and static member (object)ofclassPrintWriter,thenameofthisobjectisout.InsidePrintWriterclass,thereisa public, non-static method called println that has several overloads.

IDESoftware

IDEstandsforIntegrated DevelopmentEnvironment,IDEisasoftwarethathelpsus towrite,compile,runanddebugprograms. IDEprogramsvaryandeachoneofthemusually targeted to certain programming language, the following table shows well-known IDE programs:

Tabel1.1:ProgramminglanguagesandrespectiveIDE

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

WecannoticefromtableabovethatwehavevariousbrandsofIDEs,sinceweareinterested inJavaprogramminglanguageinthiscourse, wewillfocusonitsIDEs.NetBeansisthebest 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.

NetBeansIDE

Figure shows the main screen of NetBeans IDE (version 8.1). You can download NetBeansfromhttp://java.sun.com,onceyouhavelatestJDKversioninstalled,NetBeans installation is a straight-forward process.

TouseNetBeansforwritingandrunningyourJavaprograms, followthese steps:

1. Create a new project by going to File > New Project, or by clicking on New Project buttononthetoolbar;allprogramswrittenusingNetBeansmustbeincludedinprojects 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.
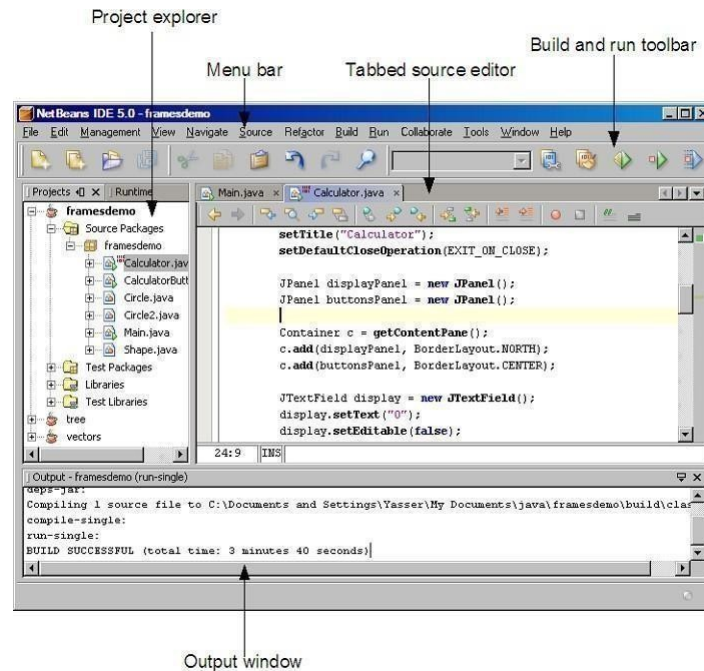
.

Figure1.10:ScreenshotofJava IDE

2. After creating a project, you are ready to start coding, on the project explorer you willseeatreethathastheprojectnameyouspecifiedasroot.Projectissplitintofour folders: Source packages, Test packages, Libraries and Test libraries. Source packages is the folder containing the source code files (.java files) and hence it isthe 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 codeeditor, 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 session1, 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 buttononthetoolbar

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

.

IntroducingObjectOriented

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 dataand associated behaviors. So, knowingwhat an object is, what does itmeanto 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 simplyspecifies what styleof software developmentis 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):

- reviewourhistory
- applyforjobs
- browse,compare,andorderproducts

In some ways, analysis is a misnomer. The baby we discussed earlierdoesn'tanalyze 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

inasinglestep,wewouldhaveturnedtherequirements defined during object-oriented analysisinto a set of classes and interfaces that could be implemented in (ideally) any object-oriented programming language.

IntroductiontoObjectOrientedProgrammingwithJava

Object-orientedprogramming(OOP)isawayofthinkingaboutandorganizingcodefor maximumreusability.Withthistypeofprogramming,aprogramcomprisesobjectsthat can interact with the user, other objects, or other programs. This makes programs more efficient and easier to understand.

Object-OrientedProgramming(OOP)isaprogrammingparadigmthatiswidelyused in Java and many other programming languages. It revolves around the concept of objects,whichareinstancesofclasses,andtheorganizationofcodeintoreusableand modularstructures.JavaisapopularlanguageforOOP,anditprovidesstrongsupport for OOP principles.

CharacteristicsofObjectOrientedLanguages.

Themajorelementsofobject-orientedlanguagesingeneralare;

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

DataTypesinJavaBasicDataTypes:

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

.

ABasicJavaProgram.

```
publicclassHelloWorld{
publicstaticvoidmain(String[]args){ S
    ystem.out.println("Hello,World!");
}
 }
```

JavaUserInput

The Scanner class is used to get user input, and it is found in the java.util package.To use theScannerclass,createanobjectoftheclassanduseanyoftheavailablemethodsfoundin the Scanner class documentation.

```
importjava.util.Scanner;//ImporttheScannerclass public

class UserInputExample {
    publicstaticvoidmain(String[]args){
        //CreateaScannerobjectttoreaduserinput
        Scannerscanner=newScanner(System.in);

        // Prompt the user for input
        System.out.println("Enteryourname:");

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

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

        //Closethescanner scanner.close();
    }
}
```

LabExercises

1. Write a Java program that prompts the user to enter their birth year and
   then calculates and displays their age.

Code:

```java
package com.mycompany.lab.assignment;
import java.util.Scanner;

public class AgeFinder {

    public static void main(String[] args) {
//        This Java program prompts the user to enter their birth year and
//        then calculates and displays their age.

//        Variables which are required
          int birthYear, age, presentYear;


//        initializing scanner object
          Scanner input = new Scanner(System.in);

//        prompts user to input their birth year and assign it to a variable
          System.out.print("Enter your birth year: ");
          birthYear = input.nextInt();

//        prompts user to input present year and assign it to a variable
          System.out.print("Enter the present year: ");
          presentYear = input.nextInt();

//        caculating age
          age = presentYear - birthYear;

//        display output
          System.out.println("Your age is " + age);

//        closing scanner object to prevent unexpected behaviours
          input.close();
    }
}
```

Output:

```
--- exec:3.1.0:exec (default-cli) @ Lab-01-assignment ---
Enter your birth year: 2005
Enter the present year: 2025
Your age is 20
```

.

2. Write a program which inputs your name and prints its length on the display screen. For input, use Scanner class.

Code:

```java
package com.mycompany.lab.assignment;
import java.util.Scanner;

public class NameLength {
    public static void main(String[] args) {

//      This java program inputs your name and prints its length on the display screen.

//      required variables
        String name;
        int length;

//      initializing scanner object
        Scanner input = new Scanner(System.in);

//      prompts user to input their name and assign it to a variable
        System.out.print("Enter your name: ");
        name = input.nextLine();

//      calculating length of name
        length = name.length();

//      display output
        System.out.println("The length of your name is: " + length);

//      closing scanner object to prevent unexpected behaviour
        input.close();
    }
}
```

Output:

```
--- exec:3.1.0:exec (default-cli) @ Lab-02-Assignment ---
Enter your name: Abdulrehman Ali
The length of your name is: 15
```

.

3. Write a program which takes weight in KGs and converts it into pounds. For input, use Scanner class.(1 Kg = 2.2 Pounds)

Code:

```java
package com.mycompany.lab.assignment;
import java.util.Scanner;


public class WeightConverter {
    public static void main(String[] args) {

//      This java program converts weight KG to pound

//      initializing variables
        double kg, pound;

//      creating scanner object
        Scanner input = new Scanner(System.in);

//      taking input of weight in kg and assign it to a variable
        System.out.print("Enter weight (KG): ");
        kg = input.nextDouble();

//      calculating weight
        pound = kg * 2.2;

//      display output
        System.out.println(kg + "KG/s is equal to " + pound + "pound/s");

//      closing scanner object to prevent unexpected behaviours
        input.close();
    }
}
```

Output:

```
--- exec:3.1.0:exec (default-cli) @ Lab-01-assignment ---
Enter weight (KG): 60
60.0KG/s is equal to 132.0pound/s
-------------------------------------------------------------------
```

.