Lab Manuals for Object Oriented Programming

Shifa Tameer-e-Millat University

شفا تعمیرِ ملّت یونیورسٹی

Lab Manual for Object-Oriented Programming

Lab 1

Development Environment and Basic Constructs in Java

Table of Contents

Lab1: Development Environment and Basic Constructs in Java

1. Introduction

You have looked at the basic characteristics of Java and the benefits of using Java over C++. The objective of this lab is to get familiar with the Java Runtime Environment to execute your first program. You will also learn and practice how to install and configure Java Development Kit (JDK) and NetBeans IDE.

2. Relevant Lecture Material

a) Revise Lecture No. 1 and 2
b) Text Book: Java: How to Program by Paul J. Deitel, Harvey M. Deitel
    1. Read pages: 11-15, 55-63
    2. Revise the object-oriented concepts.

3. Activity Time boxing

Table 1: Activity Time Boxing

Task No.	Activity Name	Activity time	Total Time
5.1	Evaluation of Design	20mins	20mins
6.2	Setting-up Path for JDK	30mins	30mins
6.2	Walkthrough Tasks	30mins	30mins
7	Practice tasks	15mins for each task	60mins
8	Evaluation Task	60mins for all assigned task	30mins

4. Objective of the experiment

- To get basic understanding of Java language characteristics
- To write simple program and learn how to compile and run it
- To get an understanding of identifying basic errors.
- To understand the command line arguments and their purpose.

5. Concept Map

This section provides you the overview of the concepts that will be discussed and implemented in this lab.

5.1 Java Development Kit

Java Development Kit commonly known as JDK is available in three different flavors targeting all the operating systems. The three different flavors are as follows

- J2SE (Standard Edition)
- J2ME (Micro Edition)
- J2EE (Enterprise Edition)

In this course, we will be studying about developing desktop applications so we will be using J2SE. J2ME is used to develop mobile applications and J2EE is used to develop server side applications.

5.2 Java Virtual Machine

A Java virtual machine (JVM)is used to execute the class files or bytecode. JVM's are available for all the operating systems and can also be configured to run on any hardware directly. Java Runtime Environment commonly known as JRE is provided by the JVM.

5.3 Java class file

The file with extension "class" that you get after compiling the Java code is called "Java class file". This is the same file which is also called the bytecode. Java class file is executed by the JVM to generate output of our program.

5.4 Class Modifiers

Java also differs from C++ in a sense that the classes now have the modifiers too. In C++, classes had modifiers i.e. public, private and protected only in case of inheritance but now in Java whenever we declare a class we now have to define its modifier as well. Following line shows the syntax for adding class modifier in "student" class

```
public class Student {
}
```

It is important to note here that any class that contains the method *main()* must be declared public.

5.5 Static functions

Static methods and static variables are the same thing. Both are the properties of the class and not the properties of the object. That means that to call a static method or static variable you will not need to create an object, you will be able to call it directly using class name.

6. Homework before Lab

There is no homework as it is the 1st lab.

7. Procedure& Tools

In this section you will study how to set up and configure JDK and NetBeans.

7.1 Tools

Java Development Kit (JDK) 19
NetBeans 17

7.2 Setting-up JDK 19 [Expected time = 30mins]

7.2.1 Download JDK

Go to www.oracle.com and search for the JDK that is compatible with your Operating system. You will probably be using windows so, download the JDK version that is available for Microsoft Windows. Figure 2 shows the download page for JDK 19.0.2
Download Link: https://www.oracle.com/java/technologies/downloads/#java19

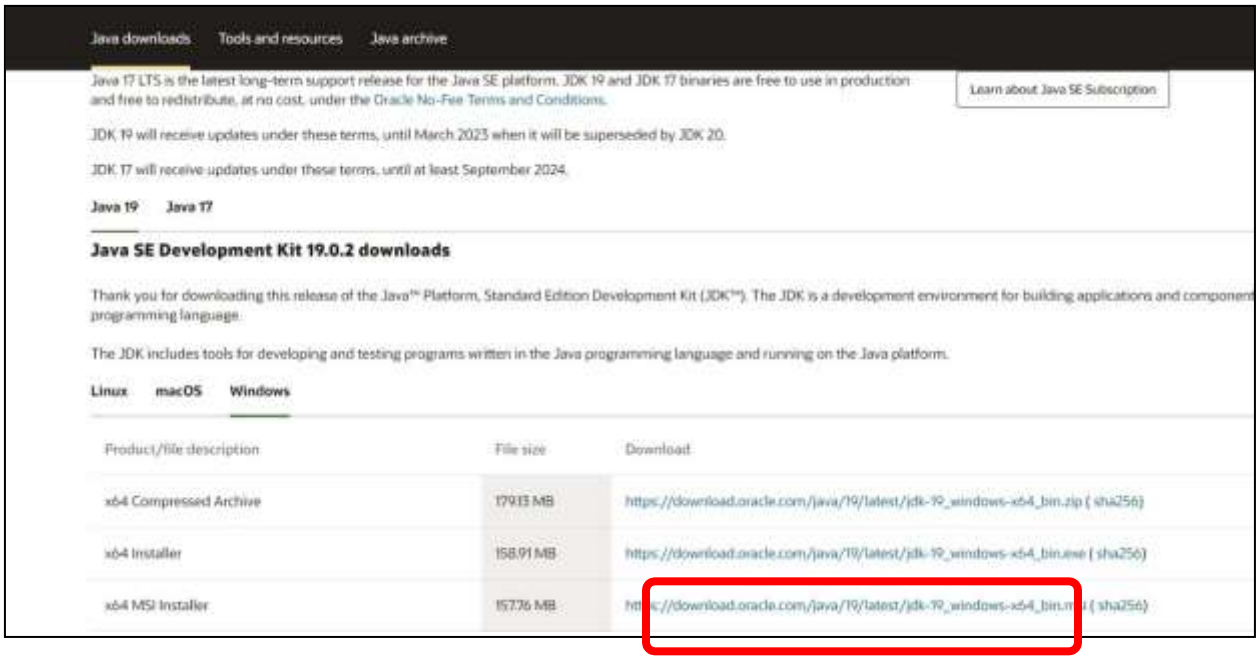

Figure 7.1: JDK19.0.2 download page

7.2.2 Install JDK

Download x64 Installer.

Install by executing the jdk-19_windows-x64_bin.exe file or whichever exe you have downloaded. **It is recommended that you do not change the paths of installation directory asthis manual is written with reference to the default paths.**

When you click the jdk exe file the following window will appear. Now click on the **Next**.

After finishing the installation of jdk next step is to set its path.

7.2.3 Set Path

Path statement enables our environment to include the new utilities. Like for newly installed JDK, the Windows environment/console does not know that there are new exe/utilities that can be used to compile and run a Java program. Therefore, it is important to set the path for new utilities using the following steps.

- Click on Start menu
- Click on Run
- Type cmd and press enter
- New console window will open
- Type ***set path= C:\Program Files\Java\jdk-19\bin***

Now your environment will be able to recognize commands like java, javac.
Remember, that you will have to set the path for each new console that you open or you will have to make bin as the current working directory. Figure 3 shows how to make bin directory as your current working directory.

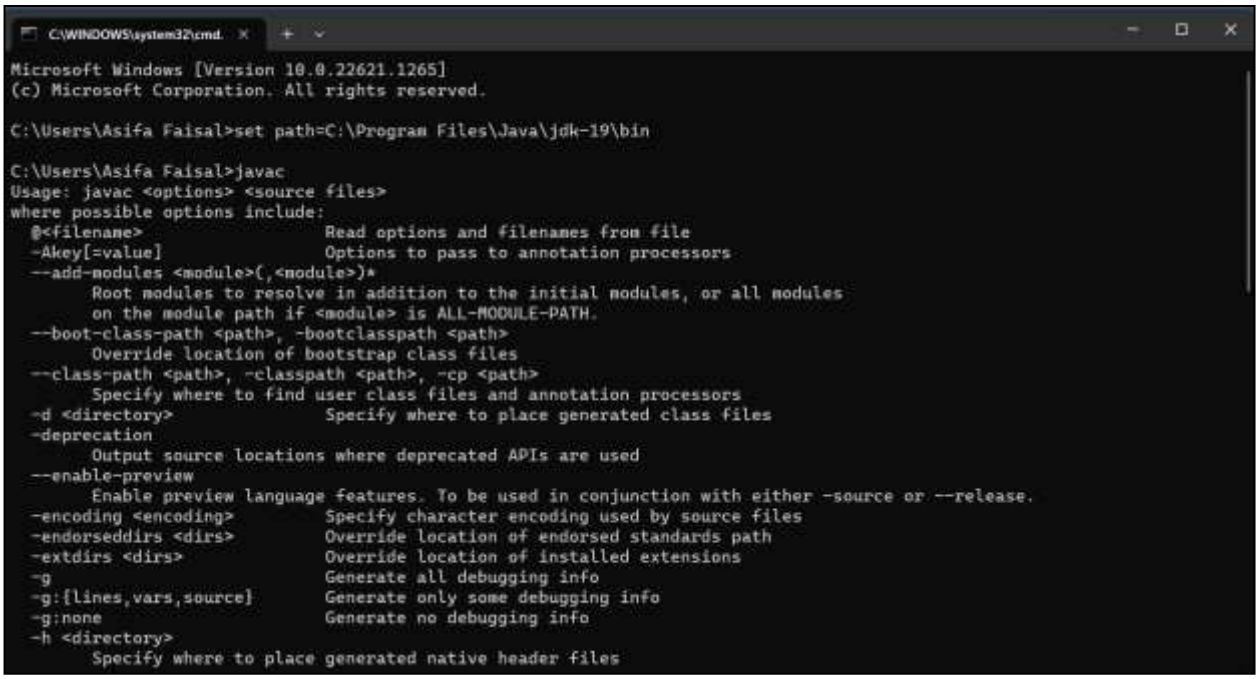

Figure 7.2: Setting bin as current working directory

7.2.4 Set Classpath

classpath in Java is a term that refers to directory or directories which are used by JVM to resolve the classes in a Java program. Classpath can be specified using CLASSPATH environment variable or –cp flag while compiling the program.
In Windows current working directory is always included in the class path. To include other directories or packages in the classpath type the following command.
Set classpath= **C:\"Directory Name"\;**

7.2.5 Set Path Permanently

As previously mentioned, that you will have to set the path every time for each new console that you open. To make things simple you can permanently set the path so it will persist after you restart the system.

To set the PATH variable permanently, add the full path of the jdk-19\bin directory to the PATH variable. Following steps will guide you through the process on Windows.

1. Click **Start**, then **Control Panel**, then **System**.
2. Click **Advanced**, then **Environment Variables**.
3. Add the location of the bin folder of the JDK installation for the PATH variable in **System Variables**. In your case you will add the following statement:

   C:\Program Files\Java\jdk-19\bin

Figure 4 shows how to set the path permanently.

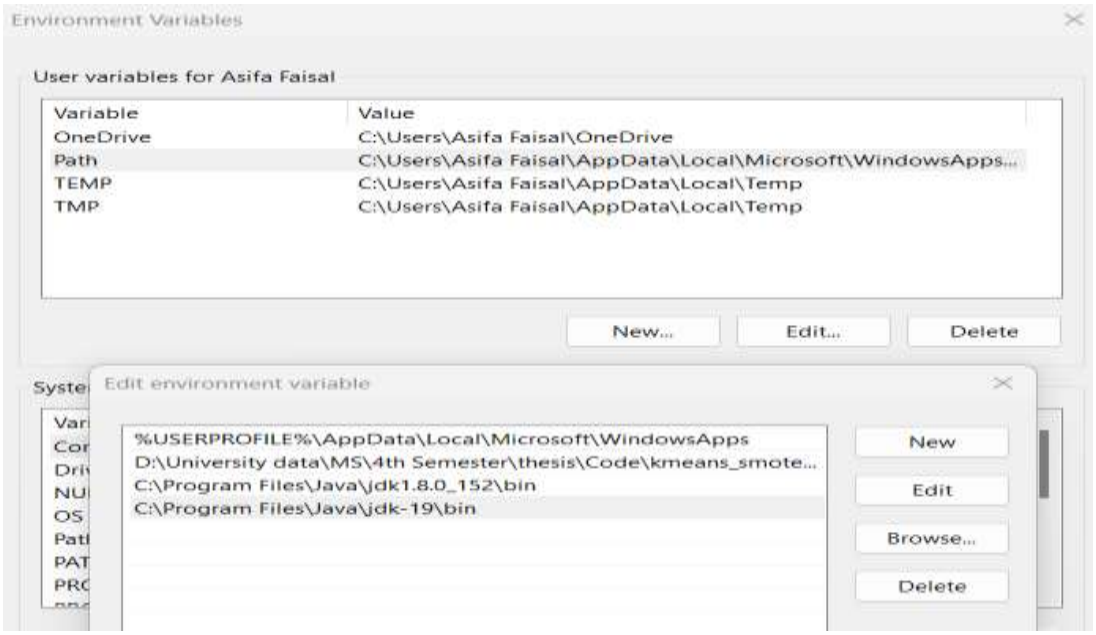

Figure 7.3: Setting Path permanently

7.2.6 Compile a Program

By now you have opened a console and completed set 6.2.4 and 6.2.5. You have also created a Hello World program described in section 1.1. Now is the time to finally compile the program. Your current working directory must contain the class named MyFirstProgram. Use the following command to compile the program.

javacMyFirstProgram.java

After the execution of the above statement bytecode of your class will be generated with same name as the .java file but its extension will be changed to .class. Now,you will need JVM to run the program.

7.2.7 Run a Program

After successfully completing the section 6.2.6, the only thing left is to execute the bytecode on the JVM. Use the following command to run the program.

javaMyFirstProgram

If the above command executed successfully then you will see “Hello World” printed on the screen.

7.3 NetBeans 17

7.3.1 Download NetBeans

Go to **https://netbeans.apache.org/** and you will see the download button here. Click the button and install the exe file named **Apache-NetBeans-17-bin-windows-x64.exe (SHA-512, PGP ASC)**

Download Link: https://www.apache.org/dyn/closer.cgi/netbeans/netbeans-installers/17/Apache-NetBeans-17-bin-windows-x64.exe

7.3.2 Install NetBeans

Install by executing the **Apache-NetBeans-17-bin-windows-x64** file or whichever exe you have downloaded.
When you click the exe file the following screen will open after configuring the installer.

Figure 5: Staring the installation of NetBeans

Click the next button then check the check box of agreement then again click the next button. You will see the following window.

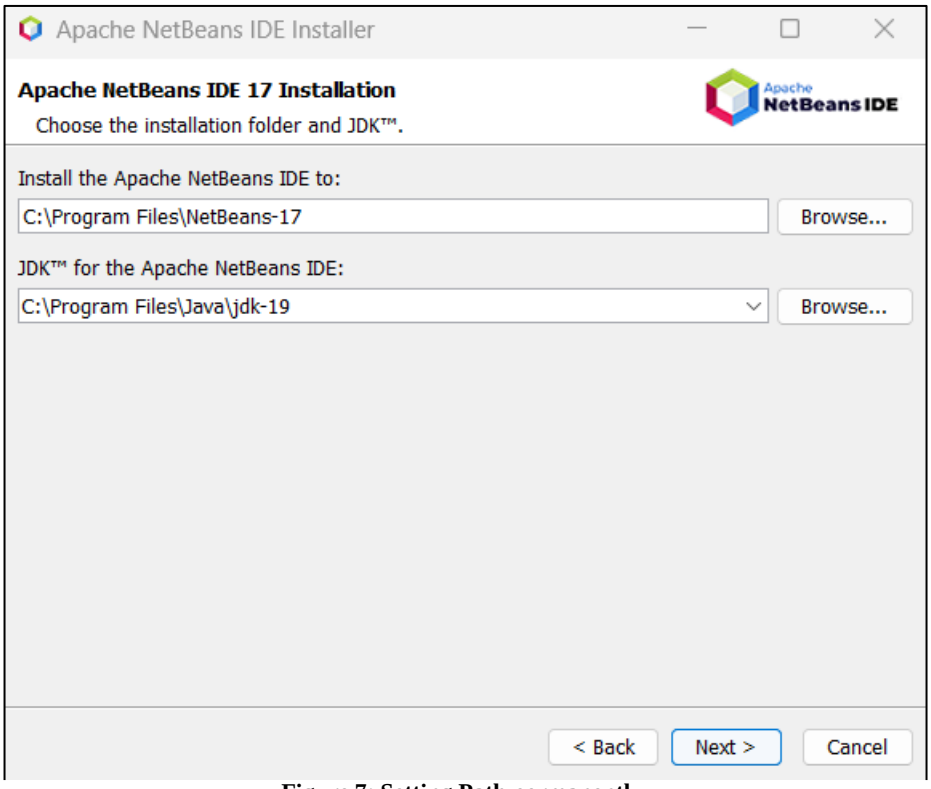

Figure 7: Setting Path permanently

Don't change the path because it is recommended to install it on its default path and simple click the next button.

After that click on the install button. The installation will start as shown in Figure 7.

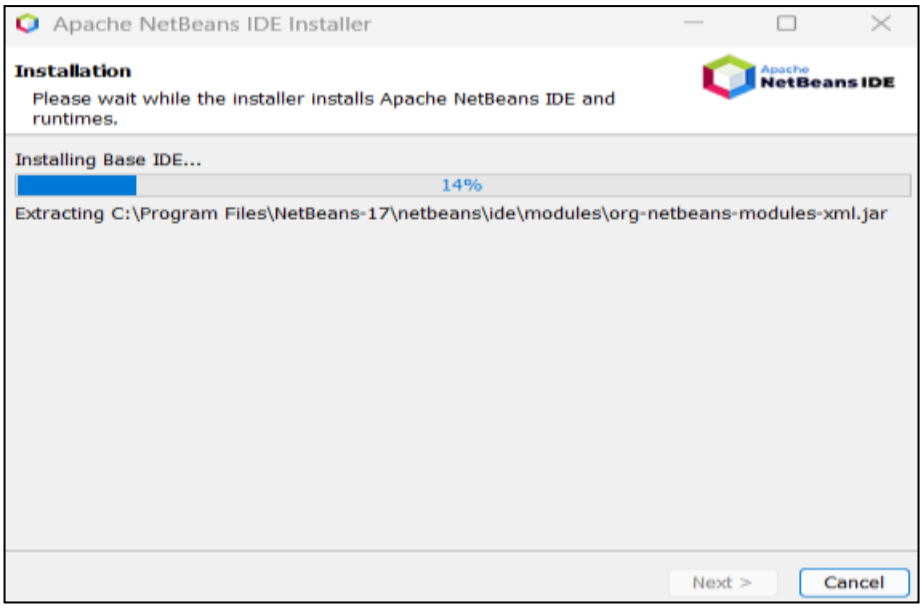

Figure 7: Installation started

After finishing the installation click the finish button.

7.3.3 Creation of project in NetBeans

To see how to create your 1st project in NetBeans click on the following link.

Creation of 1st project:
https://netbeans.apache.org/kb/docs/java/quickstart.html#:~:text=Apache%20Maven%20Website.-,Setting%20Up%20the%20Project,Then%20click%20Next.

Every application created in Java must contain a class that has a method called main() which acts as an entry point like the main() we used to define in C++. After compiling the program/ classes you will use the name of class that contains the main to start/ run your program. There are no restrictions for the name of your class, you can give any name to the class but the name of entry point function is always called main(). When you run your program, main() method will be used to call other methods from the classes in your program. For your first lab, you will only be writing a program which has only one class and only one method called main().Followingsection explains how to write your first program while highlighting the importance of all the constructs used in the program.

Writing "Hello World" program

Figure 8 shows a "Hello World" Program. You can understand the program by reading the explanations given alongside of code.

You need to enter the program code using your favorite plaintext editor e.g. a Notepad; you can change the editors at the later stages to modify the code. By writing the code in the Notepad you will easily get familiar with the naming conventions of Java.

When you have typed the code, save the file with the same name as the one used for the class and with the extension ".java". Remember that the name is case sensitive.

For this example the file name will be MyFirstProgram.java.

Example:

The program has a single class called MyFirstProgram. The class contains only one method, calledmain(). The first line of main() is always like the following

public static void main(String[] args)

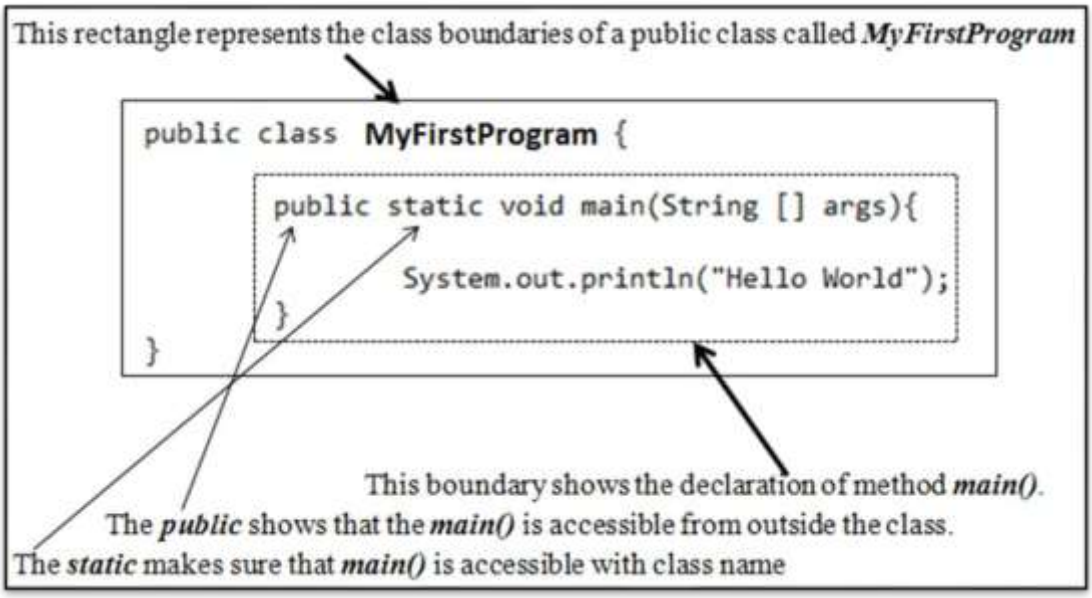

Figure 8: "Hello World" Program

System:

System is the name of the class that is contained in the package called java.lang. The package is by default included in you program. System provides you access to command line arguments and helps to display the data on the console.

out:

The method out is a static data member defined in the class called System. "out" represents the output stream.

println:

The statement println("Hello World") calls the println() method that is the property of out object. Its purpose is similar to the cout<< statement of C++ as it displays whatever is passed in the parenthesis on the console.In this case, it will display "Hello World" on the screen.

8. Practice Tasks

This section will provide more practice exercises which you need to finish during the lab. You need to finish the tasks in the required time. When you finish them, submit your task to your instructor.

8.1 Practice Task 1 [Expected time = 15mins]

Write a program that displays even numbers from 1-20.

8.2 Practice Task 2 [Expected time = 15mins]

Display first 8 numbers of Fibonacci series.

8.3 Practice Task 3 [Expected time = 15mins]

Convert the temperature from Celsius to Fahrenheit and vice versa by getting data from user using the command line arguments. Use the following formula.

°C * 9/5 + 32 = °F

8.4 Out comes

After completing this lab, student will be able to setup JDK. He/ She will also be able to compile and run basic Java programs.

8.5 Testing

This section provides you the test cases to test the working of your program. If you get the desired mentioned outputs for the given set of inputs then your program is right.

Test Cases for Practice Task-1

Sample Input	Sample Output
	2
	4
	6
	4
	8
	10
	12
	14
	16
	20

Test Cases for Practice Task-2

Sample Inputs-1	Sample Outputs-1
	0
	1
	1
	2
	3
	5
	8
	13

Test Cases for Practice Task-4

Sample Input	Sample Output
100 F to C	37.7 C

100 C to F	212 F
5 C to F	41 F

9. Evaluation Task (Unseen) [Expected time = 30mins for tasks]

The lab instructor will give you unseen task depending upon the progress of the class.

10. Evaluation criteria

The evaluation criteria for this lab will be based on the completion of the following tasks. Each task is assigned the marks percentage which will be evaluated by the instructor in the lab whether the student has finished the complete/partial task(s).

Table 3: Evaluation of the Lab

Sr. No.	Task No	Description	Marks
1	4	Problem Modeling	20
2	6	Procedures and Tools	10
3	7	Practice tasks and Testing	35
4	8	Evaluation Tasks (Unseen)	20
5		Comments	5
6		Good Programming Practices	10

11. Further Reading

This section provides the references to further polish your skills.

11.1 Books

Text Book:

- *Java: How to Program by Paul J. Deitel, Harvey M. Deitel. Eighth Edition*
- *Java Beginners Guide: http://www.oracle.com/events/global/en/java-outreach/resources/java-a-beginners-guide-1720064.pdf*