Prerequisites 1

1. Should not use IDE for compiling & running the program
2. Use any Text editor to type the following programs
3. Use javac(JDK1.8.*) for compiling the programs
4. Use Java(JRE 1.8.*) for running the programs
5. Refer javac & java usage from command line or command prompt
6. Read about classpath
7. Create directories/folders like C:\\myprojectfiles, c:\\myclassfiles, c:\\myclasses & c:\\test

Prerequisites 2

- All the below three *.java files should be in the same folder (eg: C:\\myprojectfiles)

8. Write a java class Bird.java
    1. Package name: birds
    2. Public Method(s): fly() which prints “Am flying”
9. Write another java class Parrot.java
    1. Extends: Bird
    2. Package name: child
    3. Public Method(s): speak() which prints “Am speaking”
10. Write a main class TestRunner.java.
    1. Package name: test
    2. Public Method(s): main(String[] args)
        1. Create objects for Bird & then Parrot.
        2. Using the above object instances invoke fly() & speak() methods

Exercise 1

- Working directory: Compile all the files from C:\\myprojectfiles directory in command line (cd C:\\myprojectfiles)
- Compile output folder for all the files: c:\\myclassfiles

1. Compile the Bird.java (eg: javac Bird.java. Use appropriate options & args as per the requirement)

2. Compile the Parrot.java
3. Compile the TestRunner.java
4. Run the TestRunner (eg. java pkgName.TestRunner)

Exercise 2

- Working directory: Compile all the files from C:\\test directory in command line (cd C:\\test)
- Make sure you delete all *.class files under c:\\myclassfiles
- Compile output folder for all the files: c:\\myclasses

1. Compile the Bird.java (eg: javac c:\\myprojectfiles\Bird.java. Use appropriate options & args as per the requirement)
2. Compile the Parrot.java
3. Compile the TestRunner.java
4. Run the TestRunner in command line (eg. java pkgName.TestRunner)

Exercise 3

- Working directory: Compile all the files from C:\\myprojectfiles directory in command line (cd C:\\myprojectfiles)
- Make sure you delete all *.class files under c:\\myclasses, c:\\myclassfiles

1. Compile the Bird.java & output the class files to the directory c:\\file1 (create one if not present)
2. Compile the Parrot.java & output the class files to the directory c:\\file2 (create one if not present)
3. Compile the TestRunner.java & output the class files to the directory c:\\runfile (create one if not present)
4. Run the TestRunner from (eg. java pkgName.TestRunner)

Exercise 4

- Refer how to create jar (Java Archive in command line), update jar, list files in a jar. jar executable available under jdk bin.
- Make sure you delete all *.class files under c:\\myclassfiles, c:\\file1,c:\\file2,c:\\file3

1. Using command line Create a jar file with the name myclasses.jar
2. The above jar file should contain all the class files present in directory c:\\myclasses
3. After successfully creating the above jar, delete all *.class files under c:\\myclasses & c:\\myclassfiles
4. Run the TestRunner from C:\\myprojectfiles directory using myclasses.jar
   ps: **Creating a manifest file & mentioning the main class inside that is not needed for this exercise**.

Exercise 5

- Use the same jar(myclasses.jar) created in the previous exercise.
- Make sure you delete all *.class files under c:\\myclasses, c:\\myclassfiles, c:\\file1,c:\\file2,c:\\file3

1. Add one more print "**Am flying so high**" in the fly() method of Bird.java.
2. Compile the modified Bird.java & output the class file to the directory c:\\myclasses
3. Run the TestRunner from C:\\myprojectfiles directory using **both the c:\\myclasses & myclasses.jar in the classpath**.
   a. Observe the output when myclasses.jar is first in the classpath.
   b. Observer the output when myclasses.jar is last in the classpath. You should get the new print "Am flying so high" in the console.

ps:

- No need to create manifest file & putting main class entry in the manifest file to run the jar file.
- Just keeping the jar in the class path & run the TestRunner using java -cp command.