# Step-by-Step Process to Compile and Run Java Classes with a User-Defined Package

This guide provides a step-by-step process to correctly compile and run Java classes with a user-defined package (`mypack`).

## Step 1: Create the Package Class (`A.java`)

1. Create `A.java` with the following package declaration:

package mypack;  
  
public class A {  
 public void show() {  
 System.out.println("Hello from A");  
 }  
}

2. Save `A.java` in any directory, for example, `F:\mypack\A.java`.

## Step 2: Compile the Package Class (`A.java`)

1. Open Command Prompt in the directory where `A.java` is saved (e.g., `F:\mypack`).

2. Run the following command to compile `A.java` and place the compiled class in the correct directory structure:

javac -d . A.java

3. After compilation, the directory should look like this:

F:\  
└── mypack  
 └── A.class

## Step 3: Create the Main Class (`Usepack.java`)

1. Create `Usepack.java` in the same directory (`F:\mypack\`) with the following content:

import mypack.A;  
  
public class Usepack {  
 public static void main(String[] args) {  
 A obj = new A();  
 obj.show();  
 }  
}

2. Save `Usepack.java` in the same directory (`F:\mypack`).

## Step 4: Compile the Main Class (`Usepack.java`)

1. Open Command Prompt in the directory where `Usepack.java` is saved.

2. Run the following command to compile `Usepack.java`:

javac -d . Usepack.java

3. After compilation, ensure that `Usepack.class` is generated in the same directory, with the following structure:

F:\  
├── mypack  
│ └── A.class  
└── Usepack.class

## Step 5: Run the Program

1. In the same Command Prompt, run the `Usepack` program by specifying the classpath:

java -cp . Usepack

2. The output should be:

Hello from A

## Summary of Commands:

1. Compile `A.java`:  
javac -d . A.java  
  
2. Compile `Usepack.java`:  
javac -d . Usepack.java  
  
3. Run `Usepack` Class:  
java -cp . Usepack  
The cp command is a part of the Java command line options and stands for "classpath." It is used to specify the location of user-defined classes and packages (or libraries) that are required for running a Java program. The classpath tells the Java Virtual Machine (JVM) where to look for user-defined classes and packages that are not part of the Java standard library.

### How the classpath works:

1. **Current Directory (**.**)**: If you specify . as the classpath, it includes the current directory where you are running the command, and the JVM will look for classes there.

java -cp <classpath> <classname>