

Java in Linux

The examples have been separated into two major sections: **Without Package** and **With Package**. If your Java code uses packages, go to section 2.

1. Without Package

Suppose you have these two files which should be in the same directory:

HelloWorld.java which contains:

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        Dumb dumb = new Dumb();
        System.out.println("Hello World");
        // use doit to print the message for each argument
        for (int i = 0; i < args.length; i++)
        {
            dumb.doit(args[i]);
        }
        System.exit(0);
    }
}
```

Dumb.java which contains:

```
public class Dumb
{
    String locString;
    public Dumb()
    {
        locString = "";
    }
    public void doit(String str)
    {
        System.out.println("Hello " + str);
    }
}
```

1.1 How to Compile *ClassName.java* producing *ClassName.class*?

```
$ javac ClassName.java
```

1.2 How to Run the program?

```
$ java MainClassName programCommandLineArguments
```

- The *MainClassName* must be the name of the class that has the main method.
- *programCommandLineArguments* are the command line arguments to the main method.

1.3 Example putting it all together

1. Compile each Java file producing *ClassName.class* files.

```
$ javac HelloWorld.java
```

```
$ javac Dumb.java
```

2. Execute the class containing the main passing command line arguments

```
$ java HelloWorld Fred Wilma
```

Generated Output from that program:

Hello World

Hello Fred

Hello Wilma

2. With Package

Suppose your code is part of a package named "hello". Your code should be placed in the **hello** directory:

HelloWorld.java which contains:

```
package hello;
public class HelloWorld
{
    public static void main(String[] args)
    {
        Dumb dumb = new Dumb();
        System.out.println("Hello World");
        // use doit to print the message for each argument
        for (int i = 0; i < args.length; i++)
        {
            dumb.doit(args[i]);
        }
        System.exit(0);
    }
}
```

Dumb.java which contains:

```
package hello;
public class Dumb
{
    String locString;
    public Dumb()
    {
        locString = "";
    }
    public void doit(String str)
    {
        System.out.println("Hello " + str);
    }
}
```

2.1 How to Compile *ClassName.java* producing *ClassName.class*?

You can compile the code from within the **hello** directory:

```
$ javac ClassName.java
```

Alternatively, you can compile the code from the directory above the **hello** directory:

```
$ javac hello/ClassName.java
```

In both cases, a successful compile will create *ClassName.class* in the **hello** directory.

1.2 How to Run the program?

You should run the code from the directory above the **hello** directory:

```
$ java packageName.MainClassName programCommandLineArguments
```

- *packageName* is the name of the package for that *MainClassName* and it is also the name of the directory containing its *MainClassName.class* file.

- The *MainClassName* must be the name of the class that has the main method.
- *programCommandLineArguments* are the command line arguments to the main method.

1.3 Example putting it all together

This example assumes the .java files are in the **hello** directory. The current directory is the one above hello.

1. Compile each Java file producing *ClassName.class* files.

```
$ javac hello/HelloWorld.java
```

```
$ javac hello/Dumb.java
```

2. Execute the class containing the main passing command line arguments

```
$ java hello.HelloWorld Fred Wilma
```

Generated Output from that program:

```
Hello World
```

```
Hello Fred
```

```
Hello Wilma
```