



Object-Oriented Programming (CS F213)

Module I: Object-Oriented and Java Basics

CS F213 RL 2.1: Java Basics

BITS Pilani

Dr. Pankaj Vyas

Department of Computer Science, BITS-Pilani, Pilani Campus

CS F213 RL 2.1 : Topics



- Structure of a Typical Java Program
- How to Compile and Execute a Typical Java Program

Java Program Structure



1.

<<Documentation/
Commented Section>>

2.

<<package statement>>

3.

<<import statements>>

4.

<<interface/class
definitions>>

5.

<<main() method class >>
Driver Code

Note : Every Java Program File Should have a '.java' Extension. For Example 'Sample.java', 'Demo.java' etc.

Java Program Structure:



1. Documentation / Commented section

1.

<<Documentation/
Commented Section>>

- Java Uses Three Types of Comments

1. Single Line Comments (//)
2. Multi Line Comments (/* */)
3. Documentation Comments (/** */)

// File Name Samle.java

class Box

{

private double length; /* Length of a Box */

.....

/* Method to compute the area of a Box */

public double area() { }

}// End of class Box

Java Program Structure:

2. Package Statement



2.

<<package statement>>

- Java program can have only one (**optional**) package statement
- Should be the first statement of a Java program
- Indicates that all the interfaces and classes defined in the source Java file belongs to the named package

// File Name: Sample.java

package ABC;

class A

{

}// End of class A

class B

{

}// End of class B

Java Program Structure:

3. Import Statements



3.

<<import statements>>

- Java's library classes are organized into various packages (For Example : java.lang, java.io, java.util, java.awt, javax.swing etc)
- Package → Grouping of Functionally Related Classes
- import statement is used to import either a particular class or all the classes from a particular

import java.util.*;



imports all classes from java.util package

import java.util.Date;



imports only Date class from java.util package

import java.io.*;



imports all classes from java.io package

Java Program Structure:

4. Interface/Class Definitions



4.

<<interface/class
definitions>>

- A single source '.java' file can have any number of interface and class definitions
- An interface can have only method declarations without implementations

```
// File Name: Sample.java  
class A  
{  
}  
// End of class A  
class B  
{  
}  
// End of class B  
class C  
{  
}  
// End of class C
```

```
// File Name: Sample1.java  
interface X  
{  
}  
// End of interface X  
class B  
{  
}  
// End of class B  
interface Y  
{  
}  
// End of interface Y
```

Java Program Structure:

5. main() method class / Driver class



5.

<<main() method class >>

- In order to execute a source '.java' file, it must have class with main() method with following syntax

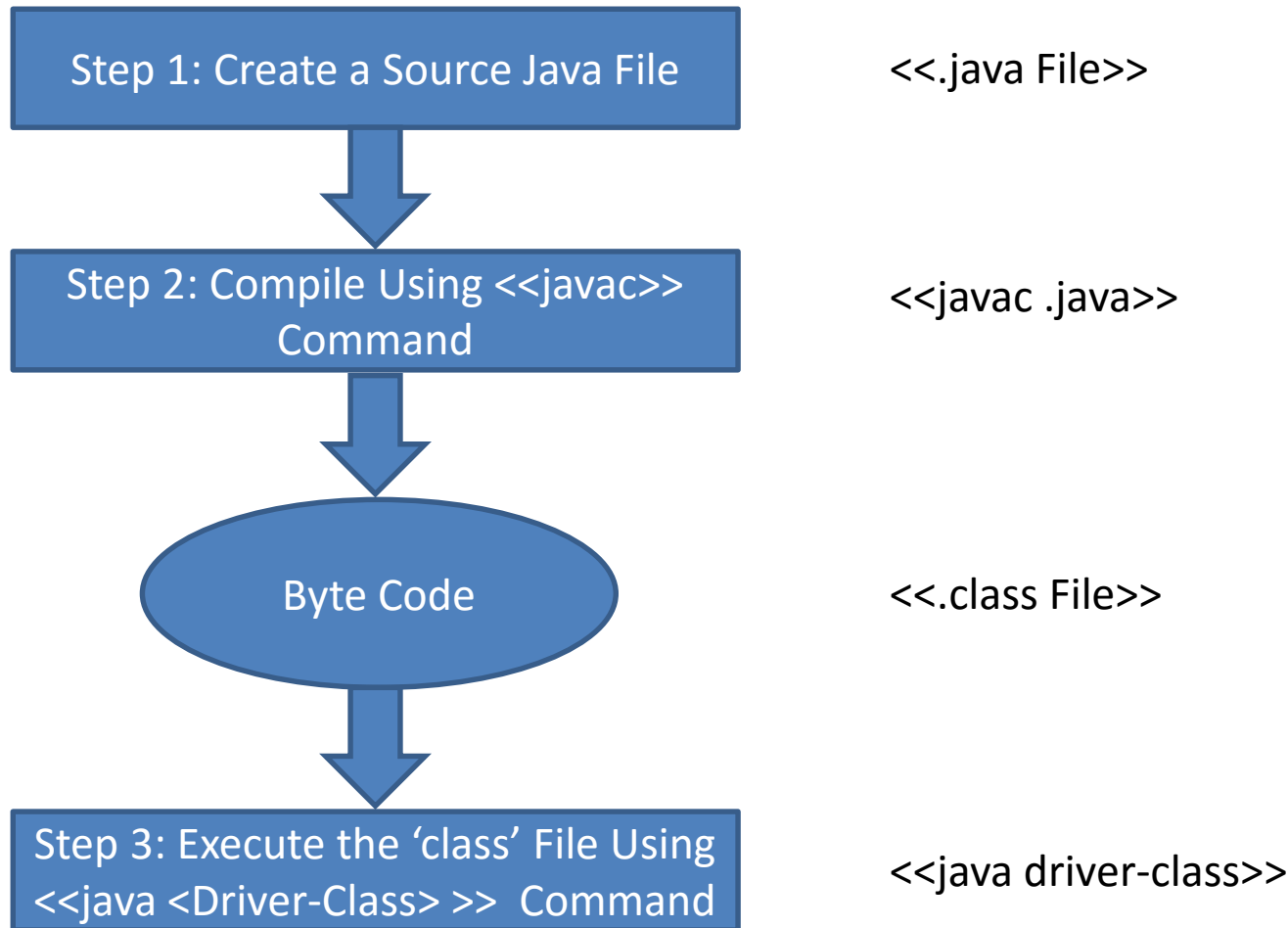
```
class X
{
    public static void main(String[] args)
    {
        // Driver main Method code
    } // End of main() Method
} // End of class X
```


Java Programming Environment



- Java's Programming Environment has two main components
 1. Java Development Kit (JDK)
[\(<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>\)](http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html)
 - Contains Java Compiler For Compiling a Java Program
 2. Java Runtime Environment (JRE)
 - Contains Java Interpreter for Executing a Java Program
- Two Famous Open Source Integrated Development Environments
 1. NetBeans (Oracle's IDE :
<https://netbeans.org/downloads/index.html>)
 2. Eclipse (<http://eclipse.org>)

Compiling and Executing a Java Program (Command Interface)



Java Program Example 1



```
// Source File Name: Example1.java
class A
{
} // End of Class A
class B
{
} // End of Class B
class C
{
} // End of class C
```

<<Compiling The Source File>>

javac Example1.java

- Each class defined in Example1.java file will be converted into .class file
- So, There three .class files will be generated for this source file named A.class, B.class and C.class
- This source file can not be executed as it does not have any driver class (class with main method)

Java Program Example 2



// Source File Name: Example2.java

```
class A
{
} // End of Class A
class B
{
    public static void main(String args[])
    {
        System.out.println("Welcome");
    } // End of main Method
} // End of class B
```

<<Compiling The Source File>>

javac Example2.java

<<Executing The Driver Class>>

java B

Welcome  Output

Java Program Example 3



// Source File Name: Example3.java

```
class A
{
} // End of Class A
class B
{
    public static void main(String args[])
    {
        System.out.println("Hello Class B");
    } // End of main Method
} // End of class B
class C
{
    public static void main(String args[])
    {
        System.out.println("Hello Class C");
    } // End of main Method
} // End of class B
```

javac Example3.java

java B

"Hello Class B"

java C

"Hello Class C"

Thank You