

File Edit Format View Help

4. Java Naming Conventions:

Java is strictly a case sensitive Programming language, where in Java applications, there is a separate recognition for lower case letters and for upper case letters.

1. All Java classes names, abstract classes names , interfaces names and enum names must be started with upper case letters and the subsequent symbols must also be upper case letters.

EX:

```
String  
StringBuffer|
```

1. All Java classes names, abstract classes names , interfaces names and enum names must be started with upper case letters and the subsequent symbols must also be upper case letters.

EX:

```
String  
StringBuffer  
InputStreamReader
```

2. All variables names must be started with lower case letters, but, the subsequent symbols must be upper case letters.

EX:

```
in, out, err  
pageContext , bodyContent  
tempEmpAddr
```

3. All Java method names must be started with lower case letters, but, the subsequent symbols must be upper case letters.

EX:

```
concat(--)  
forName()  
getInputStream()
```

4. All Java constant variables must be provided in upper case letters.

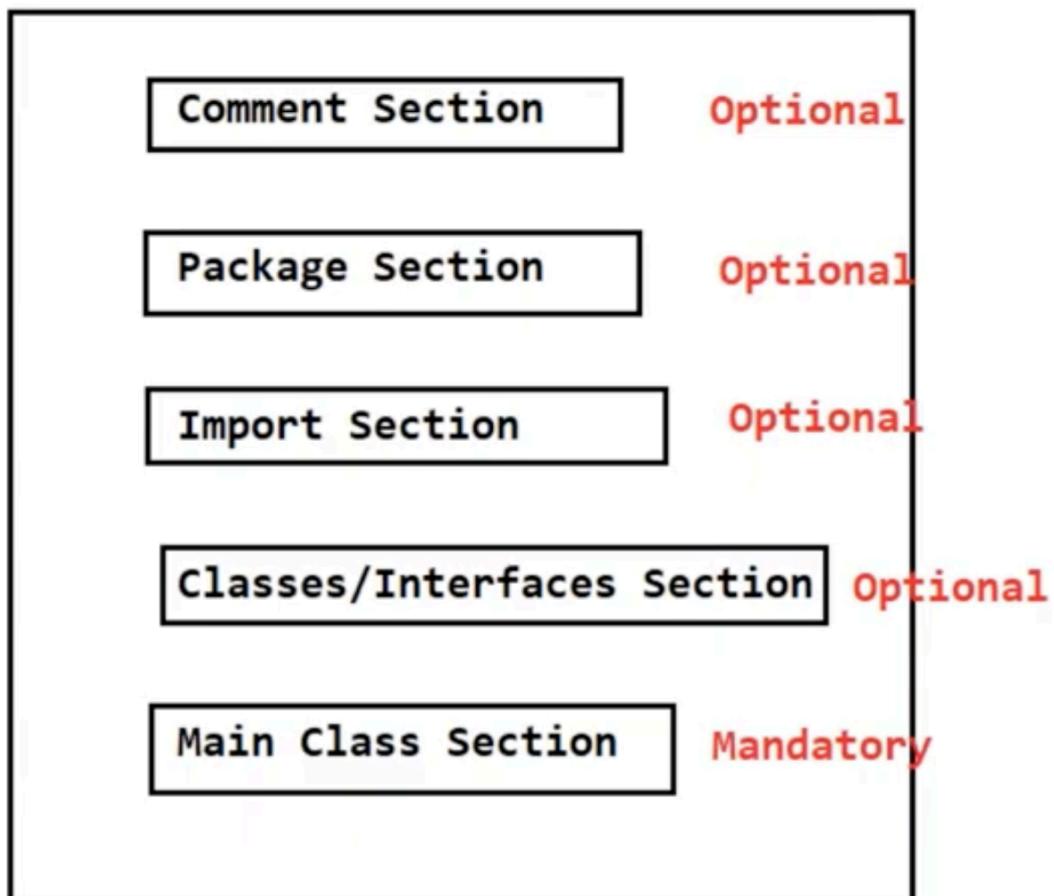
EX:

```
MIN_PRIORITY  
NORM_PRIORITY  
MAX_PRIORITY
```

5. All java packages names must be provided in lower case letters.

EX:

```
java.io  
java.util  
java.awt.event  
javax.servlet.jsp.tagext
```



5. Java Programming Format:

1. Comment Section
2. Package Section
3. import Section
4. Classes/Interfaces Section
5. main Class Section

1. Comment Section:

In Java, there are three types of comments.

1. Single Line Comments:

--> It allows the description in single line.

Syntax:

```
// --- Description---
```

2. Multi Line comments:

--> It allows description in more than one line.

Syntax:

```
/*
-----
--description-----
*/
```

```
class A{
    void add()// It perform addition operation over integer variables
    {
    }
}

class EmployeeDao
{
    /*
        Method Name: add
        Parameter List: String, String, float
        Return Type : String
    */
    public String add(String eid, String ename, float esal, String eaddr)
    {
    }
}
```

Note: To prepare API documentation we will use Documentation Comments.
API Documentation: It may be a text document or .doc file or a PDF file or an Html file, it will provide declarative information about the programming elements which we have used in the present java application.

```
Employee.java
-----
public class Employee
{
    public String eid;
    public String ename;
    public float esal;
    public String eaddr;

}
```

```
Employee.txt
-----
Class Name : Employee
Access Modifier: public
Super Class: Object
Interfaces: No Interfaces

Variables: eid, ename, esal, eaddr

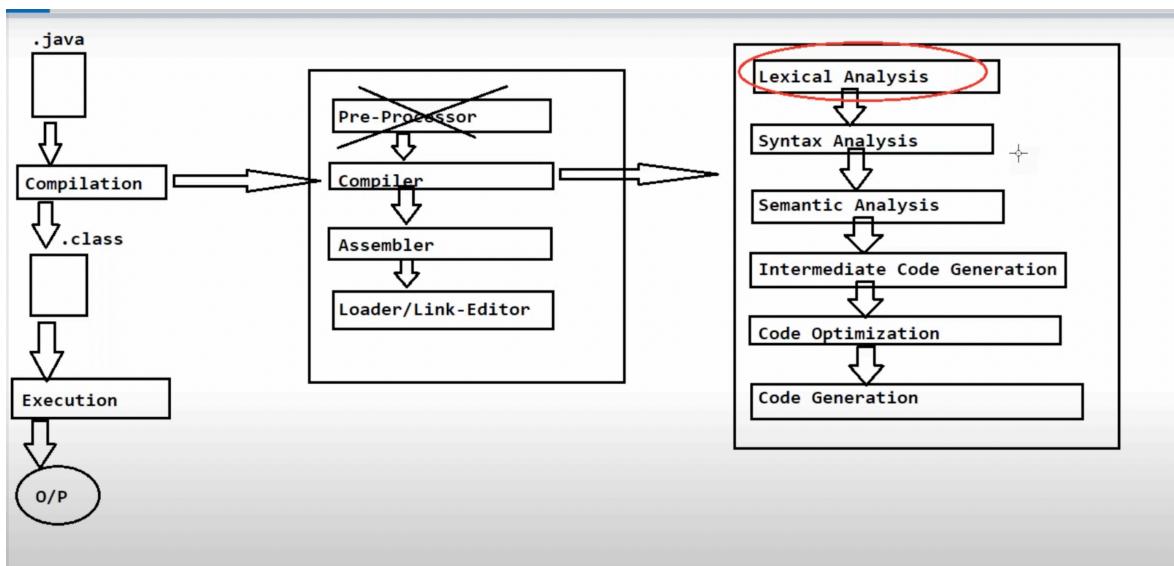
eid:
-----
Access Modifier: public
Data Type: String
Value: No value
```

```
public void add(int eno, String ename, float esal, String eaddr)
{
}
public void search(int eno)
{
}
public void update(int eno, String ename, float esal, String eaddr)
{
}
public void delete(int eno)
```

```

1  public void add(int eno, String ename, float esal, String eaddr)
2  {
3  }
4  public void search(int eno)
5  {
6  }
7  public void update(int eno, String ename, float esal, String eaddr)
8  {
9  }
0  public void delete(int eno)
1  {
2  }
3
4 }

```



```

1 public class Employee
2 {
3     public int eno;
4     public String ename;
5     public float esal;
6     public String eaddr;
7
8     public Employee(int eno)
9     {
10    }
11    public Employee(int eno, String ename)
12    {
13    }
14    public Employee(int eno, String ename, float esal)
15    {
16    }

```