**Java Naming Conventions**

The following are the key rules that must be followed by every programmer:

* The name must not contain any white spaces.
* The name should not start with special characters like & (ampersand), $ (dollar), \_ (underscore).

Class:

* It should start with the uppercase letter.
* It should be a noun such as Color, Button, System, Thread, etc.
* Use appropriate words, instead of acronyms.

Ex:

**public** **class** Employee

{

//code

}

### Interface:

* It should start with the uppercase letter.
* It should be an adjective such as Runnable, Remote, ActionListener.
* Use appropriate words, instead of acronyms.

Ex:

**interface** Printable  {

//code

}

### Method:

* It should start with lowercase letter.
* It should be a verb such as main(), print(), println().
* If the name contains multiple words, start it with a lowercase letter followed by an uppercase letter such as actionPerformed().

Ex:

**Class** Employee

{

Void draw()

{

//code

}

}

### Variable :

* It should start with a lowercase letter such as id, name.
* It should not start with the special characters like & (ampersand), $ (dollar), \_ (underscore).
* If the name contains multiple words, start it with the lowercase letter followed by an uppercase letter such as firstName, lastName.
* Avoid using one-character variables such as x, y, z.

Ex:

**class** Employee

{

//variable

**Int** id;

//code snippet

}

### Package :

* It should be a lowercase letter such as java, lang.
* If the name contains multiple words, it should be separated by dots (.) such as java.util, java.lang.

**package** com.javatpoint; //package

**class** Employee

{

//code snippet

}

### Constant :

* It should be in uppercase letters such as RED, YELLOW.
* If the name contains multiple words, it should be separated by an underscore(\_) such as MAX\_PRIORITY.
* It may contain digits but not as the first letter.

**class** Employee

{

//constant

**static** **final** **int** MIN\_AGE = 18;

//code snippet

}