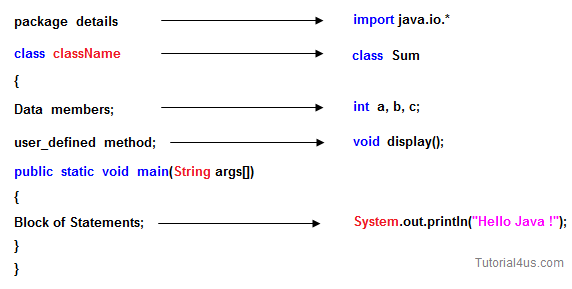
**Structure of Java Program**

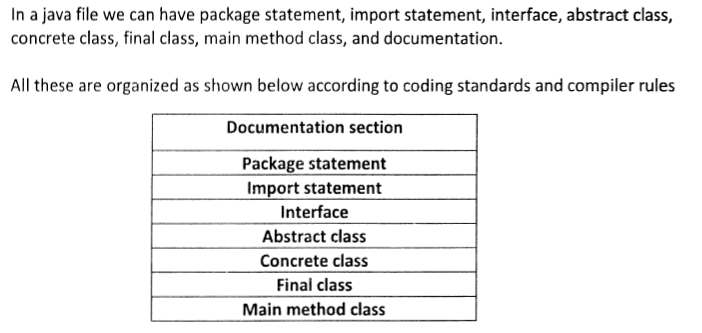
Structure of a java program is the standard format released by Language developer to the Industry programmer.

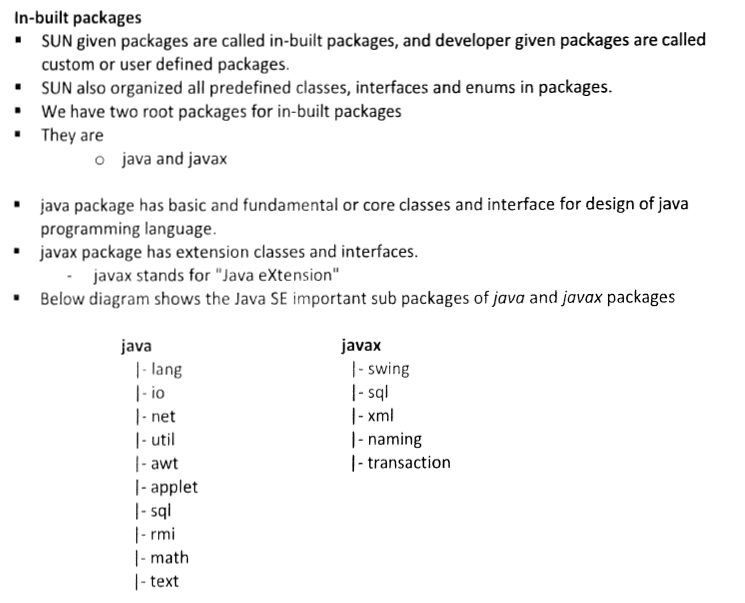
Sun Micro System has prescribed the following structure for the java programmers for developing java application.



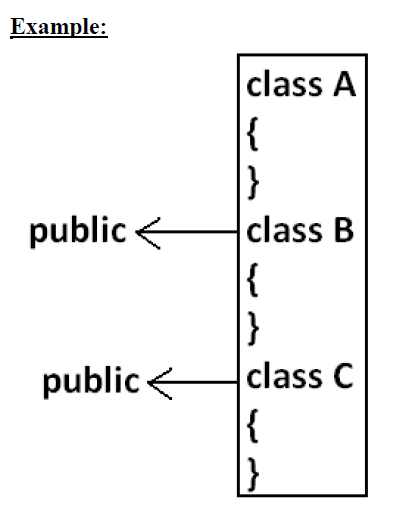
* A **package** is a collection of classes, interfaces and sub-packages. A sub package contains collection of classes, interfaces and sub-sub packages etc. java.lang.\*; package is imported by default and this package is known as default package.
* **Class** is keyword used for developing user defined data type and every java program must start with a concept of class.
* **"ClassName"** represent a java valid variable name treated as a name of the class each and every class name in java is treated as user-defined data type.
* **Data member** represents either instance or static they will be selected based on the name of the class.
* **User-defined** methods represents either instance or static they are meant for performing the operations either once or each and every time.
* Each and every java program starts execution from the main() method. And hence main() method is known as program driver.
* Since main() method of java is not returning any value and hence its return type must be void.
* Since main() method of java executes only once throughout the java program execution and hence its nature must be static.
* Since main() method must be accessed by every java programmer and hence whose access specifier must be public.
* Each and every main() method of java must take array of objects of String.
* **Block of statements** represents set of executable statements which are in term calling user-defined methods are containing business-logic.
* The file naming conversion in the java programming is that which-ever class is containing main() method, that class name must be given as a file name with an extension .java.

**Java source file structure:**

****

****

* **A java Program can contain any no. Of classes but at most one class can be declared as public. "If there is a public class the name of the Program and name f the public class must be matched otherwise we will get compile time error".**
* **If there is no public class then any name we gives for java source file.**

****

**Case 1:**

**If there is no public class then we can use any name for java source file there are no restrictions.**

**Example:**

**A.java**

**B.java**

**C.java**

**Ashok.java**

**Case 2:**

If class B declared as public then the name of the Program should be B.java otherwise we will get compile time error saying "***class B is public, should be declared in a file named B.java".***

**Case 3:**

* If both B and C classes are declared as public and name of the file is B.java then we will get compile time error saying "class C is public, should be declared in a file named C.java".
* It is highly recommended to take only one class for source file and name of the Program (file) must be same as class name. This approach improves readability and understandability of the code.

**Example:**

*class A{*

*public static void main(String args[]){*

*System.out.println("A class main method is executed");*

*}*

*}*

*class B{*

*public static void main(String args[]){*

*System.out.println("B class main method is executed");*

*}*

*}*

*class C{*

*public static void main(String args[]){*

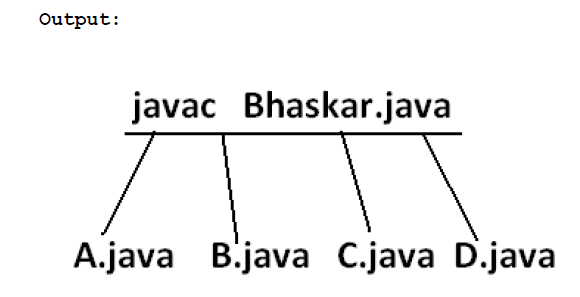
*System.out.println("C class main method is executed");*

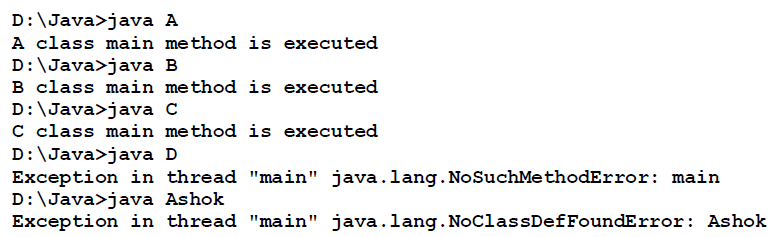
*}*

*}*

*class D{*

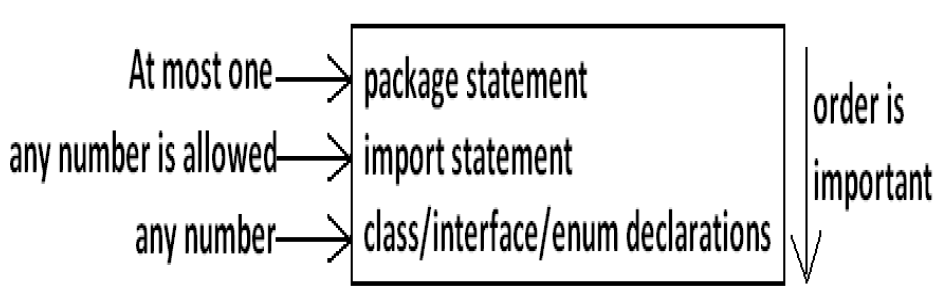
*}*

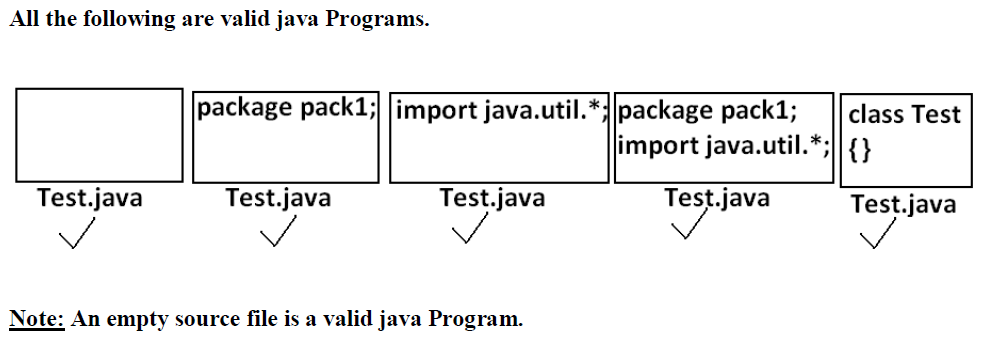
**

**

* We can compile a java Program but not java class in that Program for every class one dot class file will be created.
* We can run a java class but not java source file whenever we are trying to run a class the corresponding class main method will be executed.
* If the class won't contain main method then we will get runtime exception saying "***NoSuchMethodError: main***".
* If we are trying to execute a java class and if the corresponding .class file is not available then we will get runtime execution saying "***NoClassDefFoundError:Ashok***".

***SCJP:***

******

******

**Java coding conventions**

In Java For all predefined classes , methods, variables,...... java people already following some coding conventions and recommending us to follow same coding conventions for our own classes , methods, variables,......

**Following are different java coding conventions**

**1. Coding convention for a class**

A class name can contain any number of words but every word first letter should be in Capital letter.

**Eg:**

String, StringBuffer, InputStreamReader, VarDemo,Student,...

**2. Coding convention for interfaces**

An interface name can contain any number of words but everyword first letter should be in capital.

**Eg:**

Clonable, Runnable, Serializable, ActionListener, MyInterface,......

**3. Coding convention for methods**

A method name can contain any number of words where first word all letters should be in lowercase and second word onwards every word first letter should be in capital.

**Eg:**

main(),println(),lastIndexOf(),getAgeOfPerson(),getName(), displayData(),....

**4. Coding convention for a variables**

A variable name can contain any number of words where first word all letters should be in lowercase and second word onwards everyword first letter should be in capital.

**Eg:**

length,age,endOfTheYear,toDay,.....

**5. Coding convention for constants**

A constant name contains all letters in capital and if we have multiple words in constant name then they should separated by using underscore.

**Eg:**

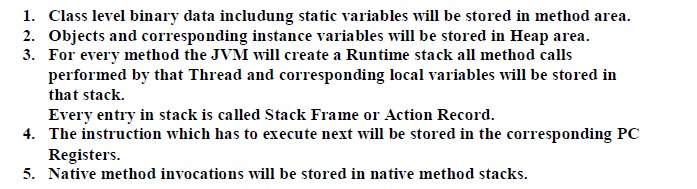
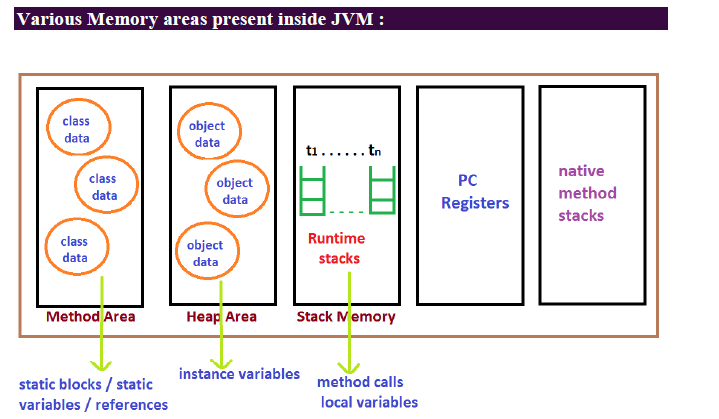
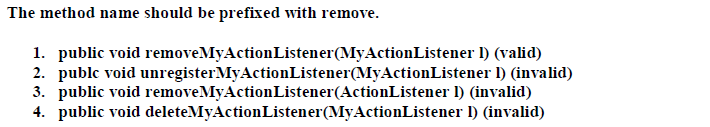
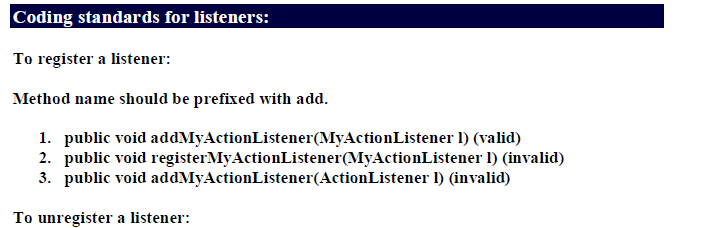
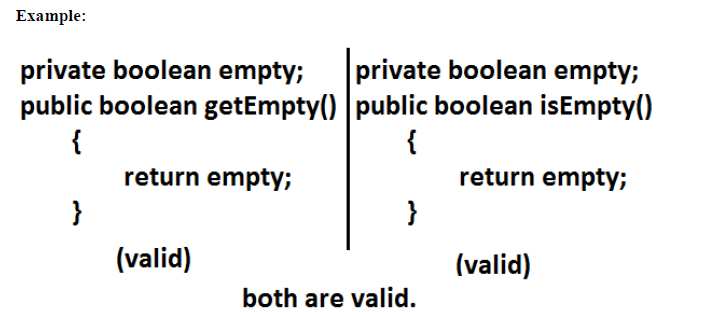
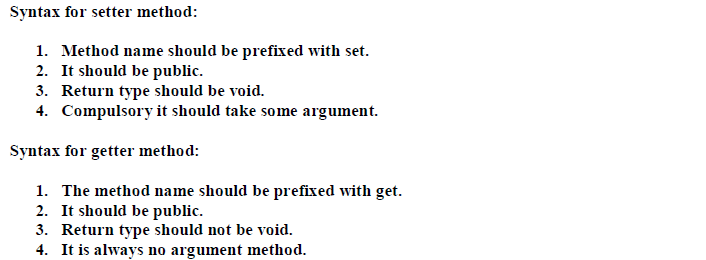
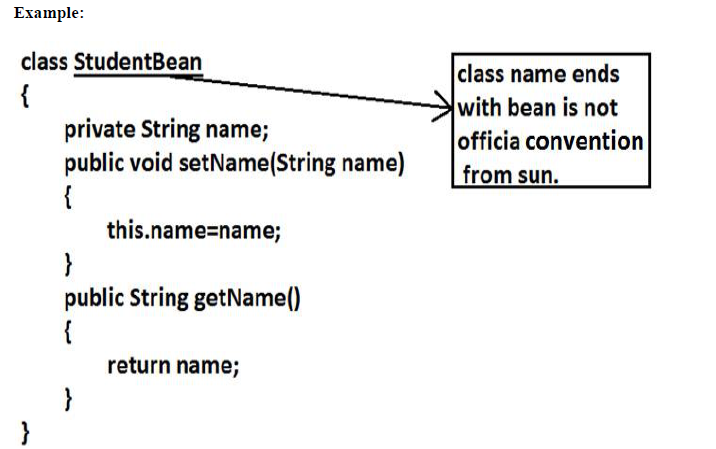
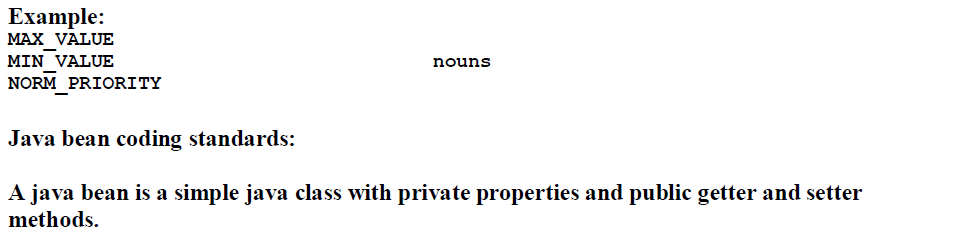
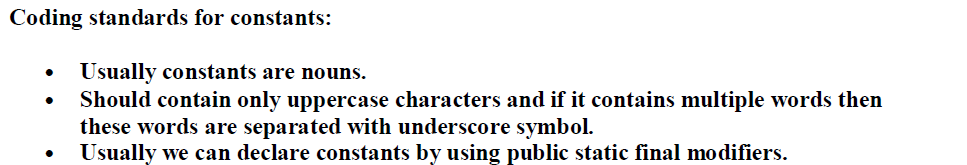
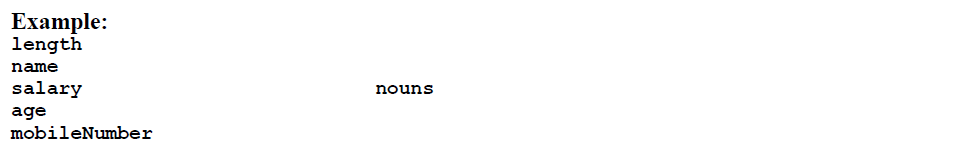
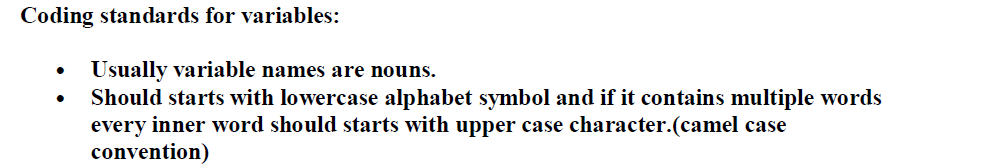
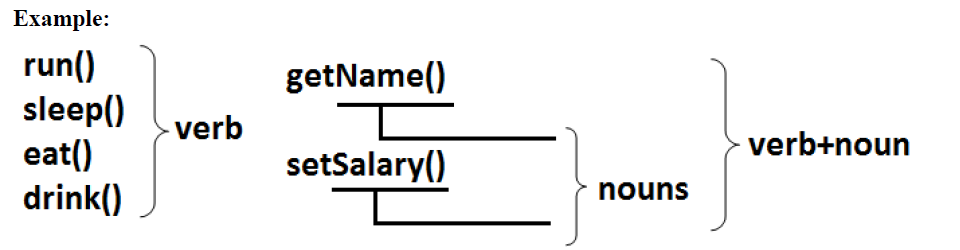
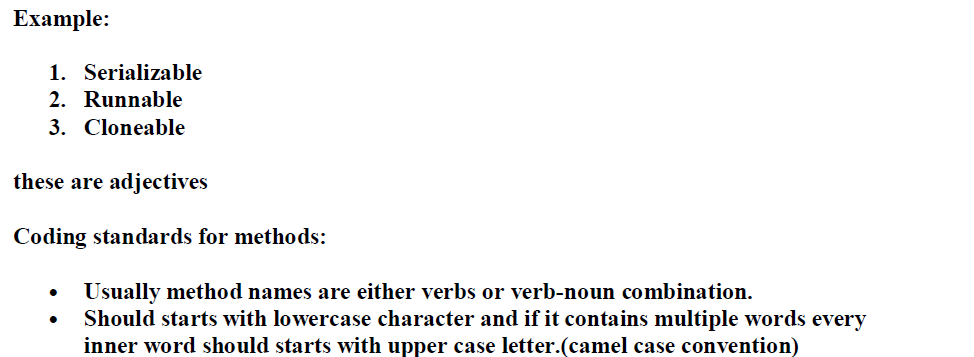
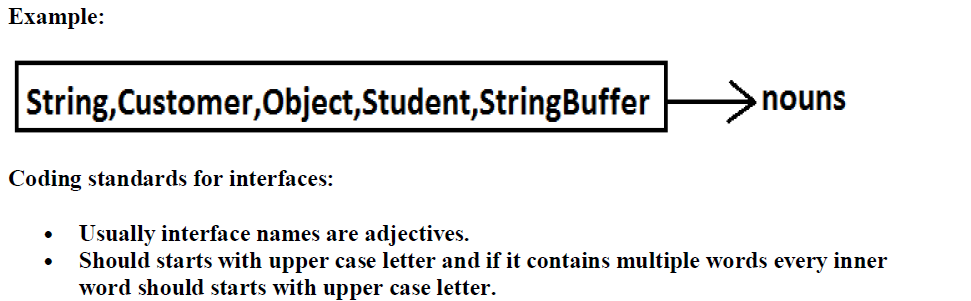
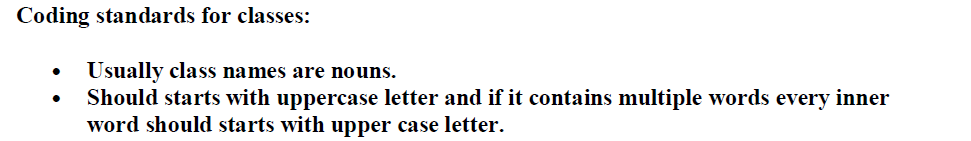
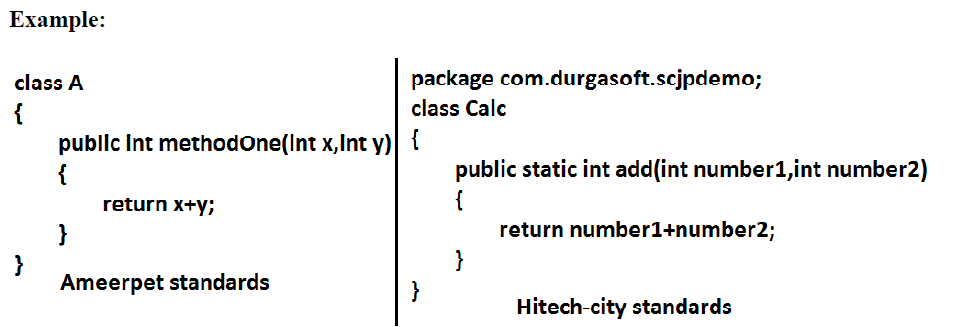
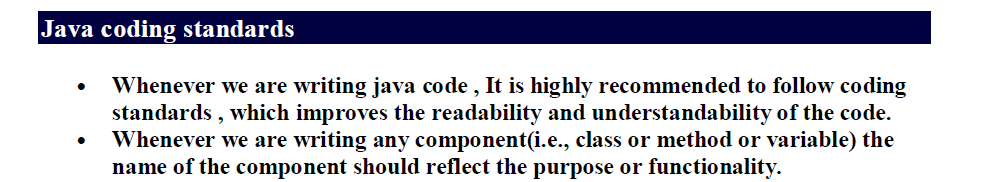
PI,MAX\_VALUE,MIN\_VALUE,EXIT\_ON\_CLOSE,MY\_CONSTANT...

**6. Coding convention for packages**

A package name contains all letters in lowercase we may or may not contain multiple words.

**Eg:**

java,lang,io,awt,mypackage,......

****

**Naming Conventions**

Naming conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier-for example, whether it's a constant, package, or class-which can be helpful in understanding the code.

|  |  |  |
| --- | --- | --- |
| **Identifier Type** | **Rules for Naming** | **Examples** |
| Packages | The prefix of a unique package name is always written in all-lowercase ASCII letters and should be one of the top-level domain names, currently com, edu, gov, mil, net, org, or one of the English two-letter codes identifying countries as specified in ISO Standard 3166, 1981.  Subsequent components of the package name vary according to an organization's own internal naming conventions. Such conventions might specify that certain directory name components be division, department, project, machine, or login names. | com.sun.eng  com.apple.quicktime.v2  edu.cmu.cs.bovik.cheese |
| Classes | Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive. Use whole words-avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML). | class Raster;  class ImageSprite; |
| Interfaces | Interface names should be capitalized like class names. | interface RasterDelegate;  interface Storing; |
| Methods | Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized. | run();  runFast();  getBackground(); |
| Variables | Except for variables, all instance, class, and class constants are in mixed case with a lowercase first letter. Internal words start with capital letters. Variable names should not start with underscore \_ or dollar sign $ characters, even though both are allowed.  Variable names should be short yet meaningful. The choice of a variable name should be mnemonic- that is, designed to indicate to the casual observer the intent of its use. One-character variable names should be avoided except for temporary "throwaway" variables. Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters. | int i;  char c;  float myWidth; |
| Constants | The names of variables declared class constants and of ANSI constants should be all uppercase with words separated by underscores ("\_"). (ANSI constants should be avoided, for ease of debugging.) | static final int MIN\_WIDTH = 4;  static final int MAX\_WIDTH = 999;  static final int GET\_THE\_CPU = 1; |