Object Oriented Programming - I

Java - Basic Syntaxes

Basic Syntax

- **Case Sensitivity** − Java is case sensitive, which means identifier Hello and hello would have different meaning in Java.

- **Class Names** − For all class names the first letter should be in Upper Case. If several words are used to form a name of the class, each inner word's first letter should be in Upper Case.
  Example: *class MyFirstJavaClass*

- **Method Names** − All method names should start with a Lower Case letter. If several words are used to form the name of the method, then each inner word's first letter should be in Upper Case.
  Example: *public void myMethodName()*

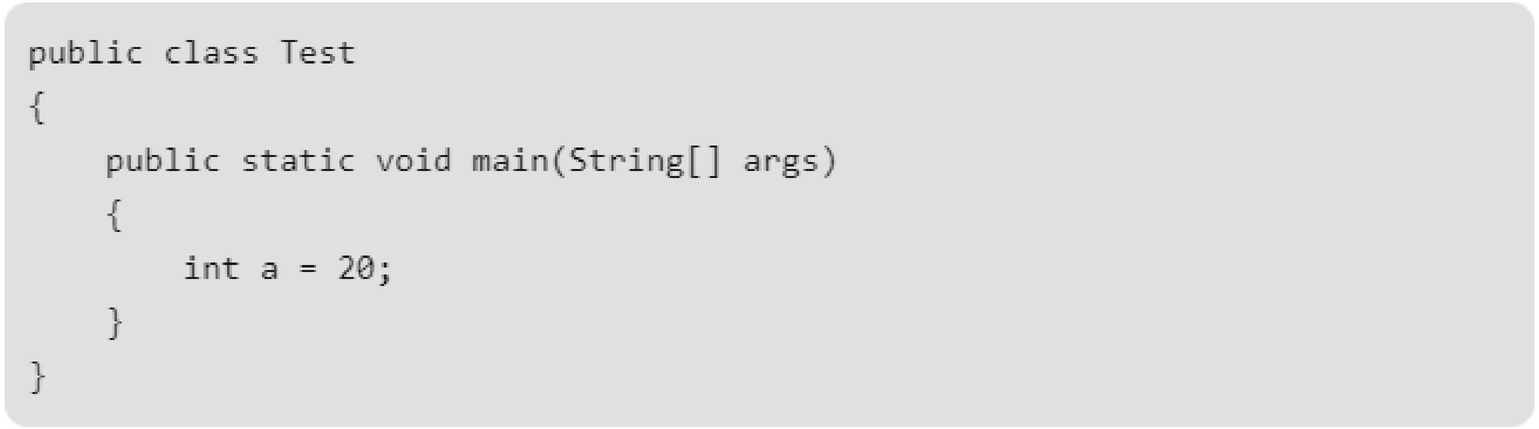
Basic Syntax

- **Program File Name** − Name of the program file should exactly match the class name. When saving the file, you should save it using the class name (Remember Java is case sensitive) and append '.java' to the end of the name (if the file name and the class name do not match, your program will not compile).
  But please make a note that in case you do not have a public class present in the file then file name can be different than class name. It is also not mandatory to have a public class in the file.
  Example: Assume 'MyFirstJavaProgram' is the class name. Then the file should be saved as *'MyFirstJavaProgram.java'*

- **public static void main(String args[])** − Java program processing starts from the main() method which is a mandatory part of every Java program.

Java Identifiers

In programming languages, identifiers are used for identification purposes. In Java, an identifier can be a class name, method name, variable name, or label. For example :



```
public class Test
{
    public static void main(String[] args)
    {
        int a = 20;
    }
}
```

Java Identifiers (Continued)

In the above java code, we have 5 identifiers namely :

- **Test** : class name.
- **main** : method name.
- **String** : predefined class name.
- **args** : variable name.
- **a** : variable name.

Rules for defining Java Identifiers

- Only allowed characters for identifiers are all alphanumeric characters ([**A-Z**],[**a-z**],[**0-9**]), ‘**$**‘(dollar sign) and ‘_‘ (underscore)
- Identifiers should not start with digits([0-9]). For example “123geeks” is a not a valid java identifier
- Java identifiers are case-sensitive
- no limit on the length of the identifier but it is advisable to use an optimum length of 4 – 15 letters only
- Reserved Words can’t be used as an identifier. For example “int while = 20;” is an invalid statement as while is a reserved word. There are 53 reserved words in Java

Valid Identifiers

MyVariable

MYVARIABLE

myvariable

x

i

x1

i1

_myvariable

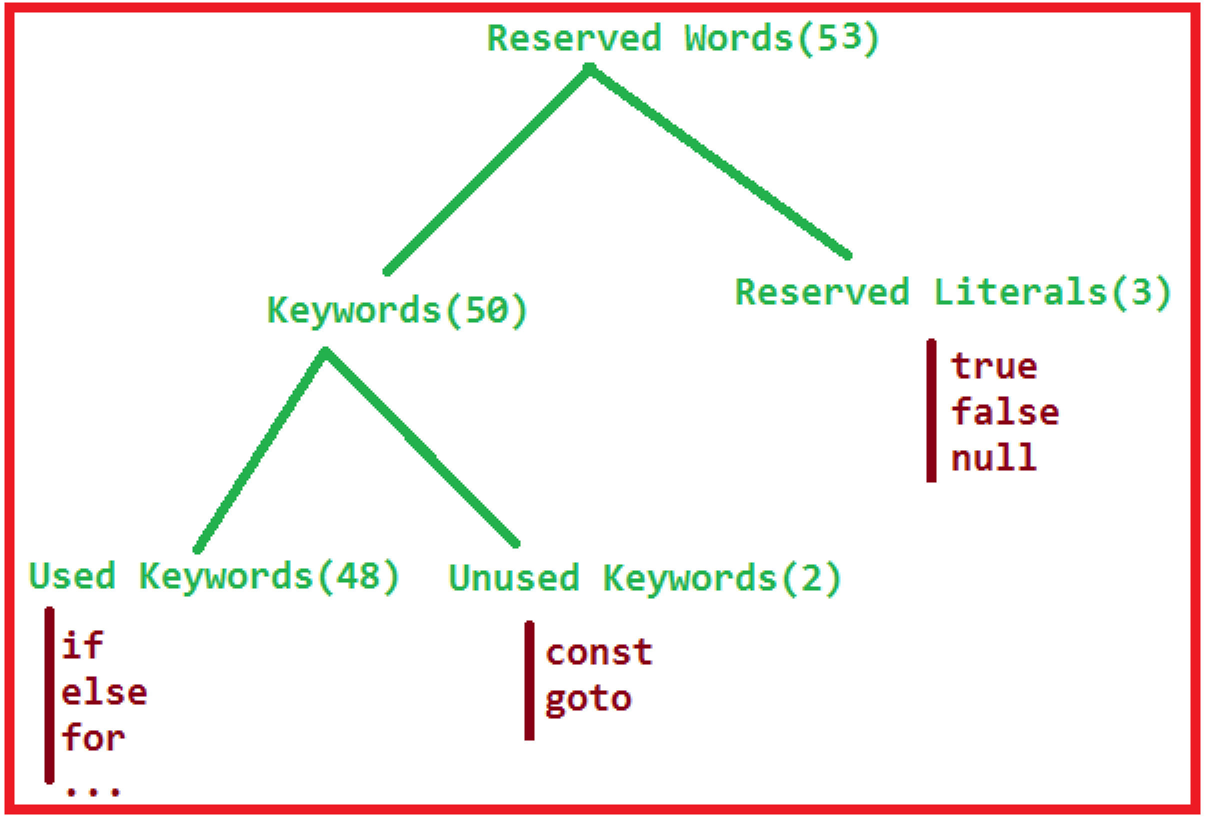
$myvariable

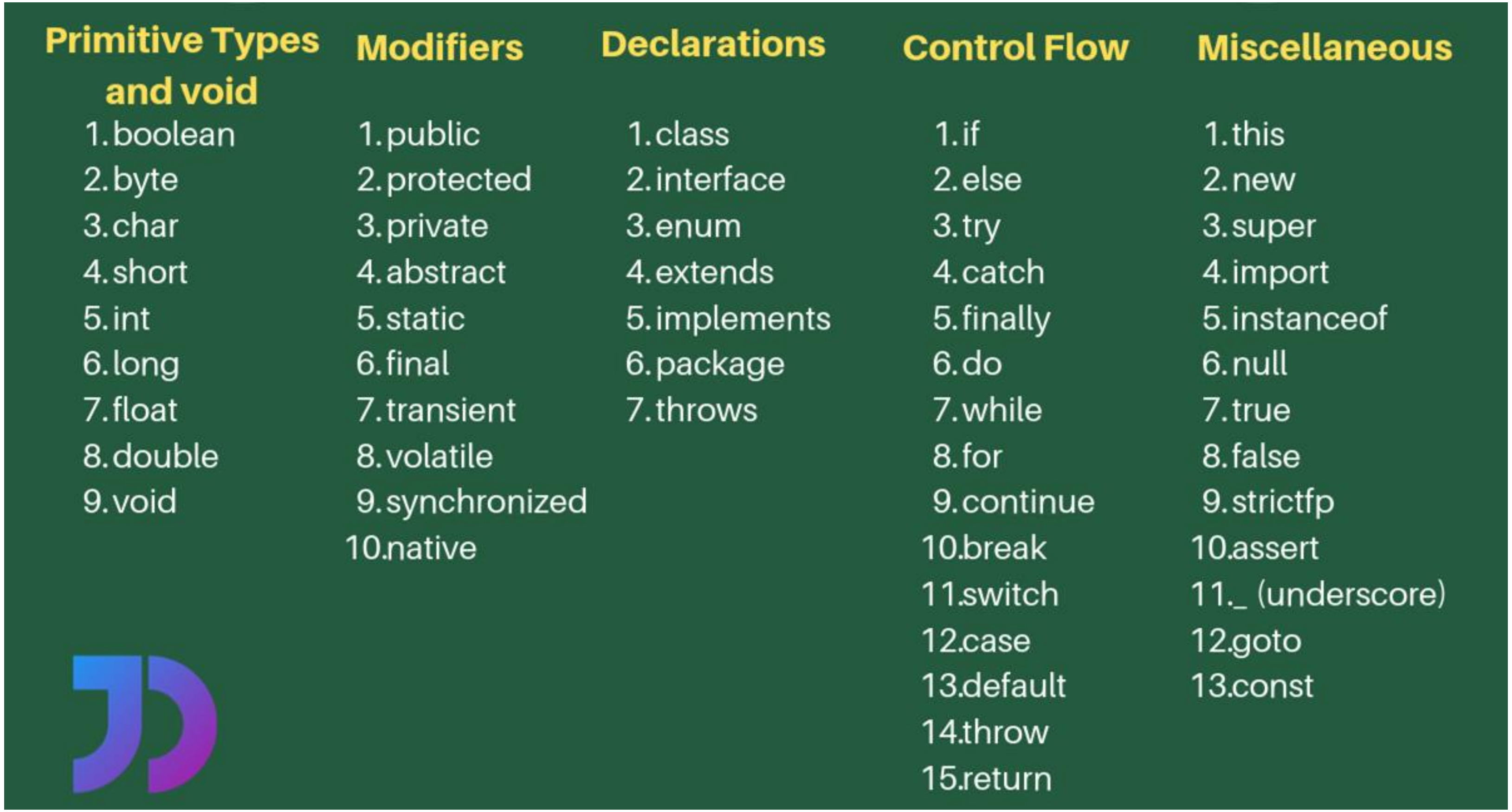
sum_of_array

geeks123

Invalid Identifiers

```
My Variable  // contains a space
123geeks   // Begins with a digit
a+c // plus sign is not an alphanumeric character
variable-2 // hyphen is not an alphanumeric character
sum_&_difference // ampersand is not an alphanumeric character
```


Reserved Words(53)
Keywords(50)
Reserved Literals(3)
true
false
null
Used Keywords(48)
Unused Keywords(2)
if
else
for
...
const
goto



Primitive Types and void	Modifiers	Declarations	Control Flow	Miscellaneous
1. boolean	1. public	1. class	1. if	1. this
2. byte	2. protected	2. interface	2. else	2. new
3. char	3. private	3. enum	3. try	3. super
4. short	4. abstract	4. extends	4. catch	4. import
5. int	5. static	5. implements	5. finally	5. instanceof
6. long	6. final	6. package	6. do	6. null
7. float	7. transient	7. throws	7. while	7. true
8. double	8. volatile		8. for	8. false
9. void	9. synchronized		9. continue	9. strictfp
	10. native		10. break	10. assert
			11. switch	11. _ (underscore)
			12. case	12. goto
			13. default	13. const
			14. throw	
			15. return	

Comments

```
public class Main {
  public static void main(String[] args) {
    // This is a comment
    System.out.println("Hello World");
  }
}
```

Comments

```
public class Main {

  public static void main(String[] args) {

    /* This is a comment This is a comment This is a comment This is a
comment This is a comment */

    System.out.println("Hello World");

  }

}
```

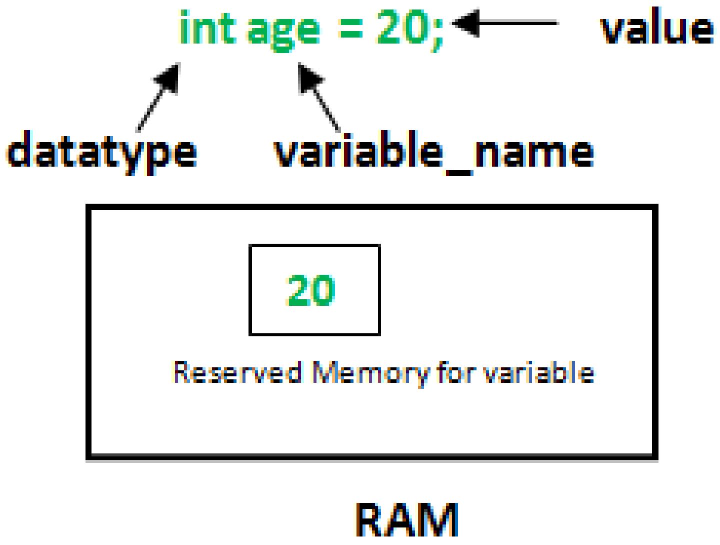
Java Variables

Variables are containers for storing data values.

In Java, there are different types of variables, for example:

- String - stores text, such as "Hello". String values are surrounded by double quotes
- int - stores integers (whole numbers), without decimals, such as 123 or -123
- float - stores floating point numbers, with decimals, such as 19.99 or -19.99
- char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- boolean - stores values with two states: true or false

Java Variables


int age = 20;
value
datatype
variable_name
20
Reserved Memory for variable
RAM