* ***What is Identifier?***

*A name in java program is called Identifier which can be used for Identification purpose. It can be method name or variable name, class name, label name.*

* ***Rules for defining Java Identifiers:-***

1. *Allowed characters in Java Identifiers are* 🡪 *a-z,A-Z, 0-9, $,\_*
2. *If we are using any other character we will get compile time error.*
3. *Identifiers cannot start with digits, special characters.*
4. *Identifiers are Case-Sensitive, language itself is created as Case-Sensitive language*
5. *There is no length limit for Java identifiers but not recommended to take too lengthy identifiers.*
6. *We can’t use reserved words/ keywords as Identifiers.*
7. *All pre-defined Java class names and interface names can be used as Identifiers, but it is not a good coding practice.*

*Example:-*

*int String=888; int Runnable=1000;*

*SOP(String); SOP(Runnable);*

*(Valid) (Valid)*

*--------------------------------------------------------------------------------------------------------------------------------------*

* ***Reserved words -53***

*In Java some words are reserved to represent some meaning or functionality, such type of words are called reserved words.*

*🡪 Keywords(50)*

*🡪 Used Keywords(48)*

*🡪 DataType Keywords(8) 🡪 byte, short,int, long, float, double, char, Boolean*

*🡪 Flow Control Keywords(11) 🡪if, else, switch,case, default, while, do, for, break, continue, return*

*🡪Modifier Keywords (11) 🡪public, private, protected, static, final, abstract, native, synchronized, strictfp (v1.2), transient, volatile.*

*🡪Exception keywords(6)*🡪*try, catch,finally, throw, throws, assert (v1.4)*

*🡪Class related keywords(6) 🡪class, interface, extends, implements, package, import.*

*🡪Object related keywords(4) 🡪new, instanceof, super, this*

*🡪 Void return type keyword (1) 🡪 void*

*🡪 static,enum (v1.5)*

*We can use enum to define named constants* ***eg, Month names- jan,feb,mar,***

*🡪 Unused Keywords(2) 🡪goto, const.*

*🡪 Reserved Literals(3) 🡪 true, false, null*

* *All 53 reserved words in java contains Lowercase Alphabet symbols.*
* *In java we have only new keyword and no delete keyword because destruction of useless objects is responsibility of Garbage Collector*

***Data Types:-***

1. *In java every variable and expression has some type*
2. *Each and every data type is clearly defined*
3. *Every assignment gets checked by compiler for type compatibility.*

*Due to above reason we can conclude java is strongly typed Language.*

***Is JAVA purely object oriented programming language?***

* *Java is not considered as pure object oriented programming language because, several OOP features are not satisfied by JAVA, like multiple inheritance, operator Overloading.*
* *Moreover we are depending on Primitive data types which are not objects.*

|  |  |  |  |
| --- | --- | --- | --- |
| *Primitive Data Types* | | | *Non Primitive Data Types* |
| *Integral (Numeric)* | *Floating,decimal* | *Non-Numeric* |  |
| ***byte 🡪0***  *(1 Byte-8 Bits)*  ***Short 🡪0***  *(2 byte- 16 bits)*  ***Int🡪0***  *(4byte – 32 bits)*  ***long🡪0L***  *(8 byte – 64 bits)* | ***Float 🡪0.0f***  *(4byte-32 bits)*  *(1.7\*10^38)*  *(6 to7 no after .)*  ***Double🡪0.0d***  *(8byte-64bits)*  *(-3.4\*10^38)*  *(15 num after .)* | ***char 🡪\u0000***  *(2 byte- 16 bits)*  *(a-z,A-Z, special characters)*  ***boolean 🡪false***  *(true, false)*  *1 Bit* |

* *Except boolean and char all datatypes are signed DT. Because we can represent them in +,-*
* ***Byte*** *is the best choice if we want to handle the data , in terms of streams, either from the file or from the network.(File supported form, Network supported form is Byte)*
* ***Short*** *is the most rarely used DT in java. It is best suitable for 16-bit processors like 8085. But these are completely outdated processor.*
* ***Float*** *provides Single precision.*
* ***Double*** *provides Double precision*
* ***Char*** *Old Languages are ASCII code based and no of allowed different characters are <=256.To represent these 256 characters 8 bits are enough , hence the size of char in old language is 1Byte. But Java is UNICODE based and no different Unicode characters are more than 256i.e 65536. To represent this much chars 8 bits not enough so need to go for 16 bits. Hence the size of char is 2 Bytes. (0 to 65535)*