Keywords, Identifiers, Data Types and Variables

Keywords

Every programming language has a set of keywords which are reserved words having a predefined meaning. Each keyword represents a specific functionality in the language.

Some java keywords are:

Keywords		
abstract	else	return
break	final	static
case	for	switch
class	if	throw
continue	import	try
default	interface	this
do	new	while

Variables

Data is stored in variables. A variable is a named memory location which holds some value. The value stored in a variable can vary during the execution of the program.

Note: In Java, it is necessary to declare a variable along with a data type before it can be used.

Variable declaration can be done in two ways:

- Declaration only
  < data type > < variable name >
- Declaration and Initialization
  < data type > < variable name > = < value >

Eg: int data = 1000;

Identifiers

An identifier is the name given to a variable, method or class to uniquely identify it.
Rules for identifiers:

1. It can contain alphanumeric characters([a-z], [A-Z], [0-9]), dollar sign ($), underscore (_)
2. It should not start with a digit ([0-9])
3. It should not have spaces
4. It should not be a Java keyword
5. It is case-sensitive
6. It has no length restrictions

Examples of valid identifiers:

var, _var, My_var, var123, $myVar

Examples of invalid identifiers:

123var, My variable, var-1

Data Types

Data type defines the type of data that can be stored in the variable and the memory required by it.

There are two types of data types in Java:

1. Primitive data types
2. Non-primitive data types

In java, primitive data types have different sizes. Primitive data types are the basic data types defined in Java. There are 8 primitive data types.

Data type	Description	Default	Size	Range	Example
boolean	stores true or false	false	1 bit	true or false	boolean v = true;
byte	stores whole numbers and is used for small values	0	8 bits	-2^7 to 2^7-1	byte a = 10;
char	stores a single Unicode character	\u0000	16 bits	'\u0000' to '\uffff'	char c = 'A';
short	stores whole numbers and is used for smaller values	0	16 bits	-2^{15} to $2^{15}-1$	short s = 10;
int	stores whole numbers and is the preferred data type for numeric values	0	32 bits	-2^{31} to $2^{31}-1$	int x = 1000;
long	stores whole numbers and is used for large values that int cannot store	0	64 bits	-2^{63} to $2^{63}-1$	long a = 100000L;
float	stores fractional numbers with 6-7 decimal digits	0.0	32 bits	1.4e-045 to 3.4e+038	float f = 23.5f;
double	stores fractional numbers with up to 15 decimal digits	0.0	64 bits	4.9e-324 to 1.8e+308	double d = 32.5;

Important points for Primitive data types:

1. Numeric and boolean (true, false) values are written without quotes. E.g. int score = 85; boolean isQualified = true;
2. The character value must be written in single quotes while assigning it to a character variable. E.g. char gender = 'M';
3. A long value is assigned to the variable, suffixed with L (uppercase letter or lower case letter L can be used). E.g. long salary = 500000L;
4. A float value must be suffixed with F or f while assigning to the variable. E.g. float average = 78.6f;

Non-primitive data types include classes, arrays, interfaces etc. These can be predefined in the Java library or user-defined in the programs. These will be covered in detail later in the course.

String is a predefined class in Java library which is used to store a sequence of characters:

```
String myName = "Eren Yeager";
```

Coding Standards (nomenclature)

1. Variable names must be nouns, starting with lowercase letters. If it contains multiple words, then every inner word must start with capital letter. This type of casing is called camel casing.
   (mobileNumber, yourAddress, paymentMode)

2. Comments:

   Comments are those statements which are not executed by the compiler. Comments can be used to provide information about a variables or statement.

   There are two types of comments in Java:

   2.1. Single Line comment

   //This is a single line comment

   2.2. Multi line comment

   /*This is a

   multi-line comment*/