JAVA LANGUAGE FUNDAMENTALS

ChandraSekhar(CS) Baratam

IDENTIFIERS

- Identifiers are the names provided by the Java programmer. These names can be assigned to variables, methods, classes, etc.
- Java identifier can be with any length.
- It Can start with a Unicode letter, underscore (_), or dollar sign ($).

JAVA KEYWORDS

- KEYWORDS ARE WORDS RESERVED FOR JAVA AND CANNOT BE USED AS IDENTIFIERS OR VARIABLE NAMES.

Java Keywords				
abstract	boolean	break	byte	case
catch	char	class	continue	default
do	double	else	extends	false
final	finally	float	for	if
implements	import	instanceof	int	interface
long	native	new	null	package
private	protected	public	return	short
static	super	switch	synchronized	this
throw	throws	transient	true	try
void	volatile	while		
Keywords that are reserved but not used by Java				
const	goto			

DATA TYPES

- The type of value that a variable can hold is called data type.
- When we declare a variable we need to specify the type of value it will hold along with the name of the variable.
- This tells the compiler that the particular variable will hold certain amount of memory to store values.

Primitives Data types

Type	Size in bits	Values
boolean	8	true or false
char	16	\u0000 - \uFFFF
byte	8	-128 - 127
short	16	-32,768 – 32,767
int	32	-2,147,483,648 - +2,147,483,647
long	64	-9,223,372,036,854,775,808 - +9,223,372,036,854,775,807
float	32	-3.40292347E+38 - -+3.40292347E+38
double	64	-1.79769313486231570E+308 to -+1.79769313486231570E+308

Variable Type	Default Value
Object reference	`null` (not referencing any object)
`byte`, `short`, `int`, `long`	`0`
`float`, `double`	`0.0`
`boolean`	`false`
`char`	`'\u0000'`

Variables Definition

- Variables are used to represent the data that a program deals with
- As the name might imply, the data that a variable holds can change.
- We have to define a variable before using it
- Here is an example of defining a variable:

int number; **float b;**

TYPES OF VARIABLES

1. Local Variables:

- The variables defined in a method or block of code is called local variables.
- These variables can be accessed within a method or block of code only.
- These variables don't take default values if not initialized.
- These values are required to be initialized before using them.

TYPES OF VARIABLES CONT...

2. Instance Variables (Non-static fields):

- In object oriented programming, objects store their individual states in the "non-static fields" that is declared without the static keyword.
- Each object of the class has its own set of values for these non-static variables so we can say that these are related to objects (instances of the class).
- Hence these variables are also known as instance variables. These variables take default values if not initialized.

TYPES OF VARIABLES CONT...

3. Class Variables (Static fields):

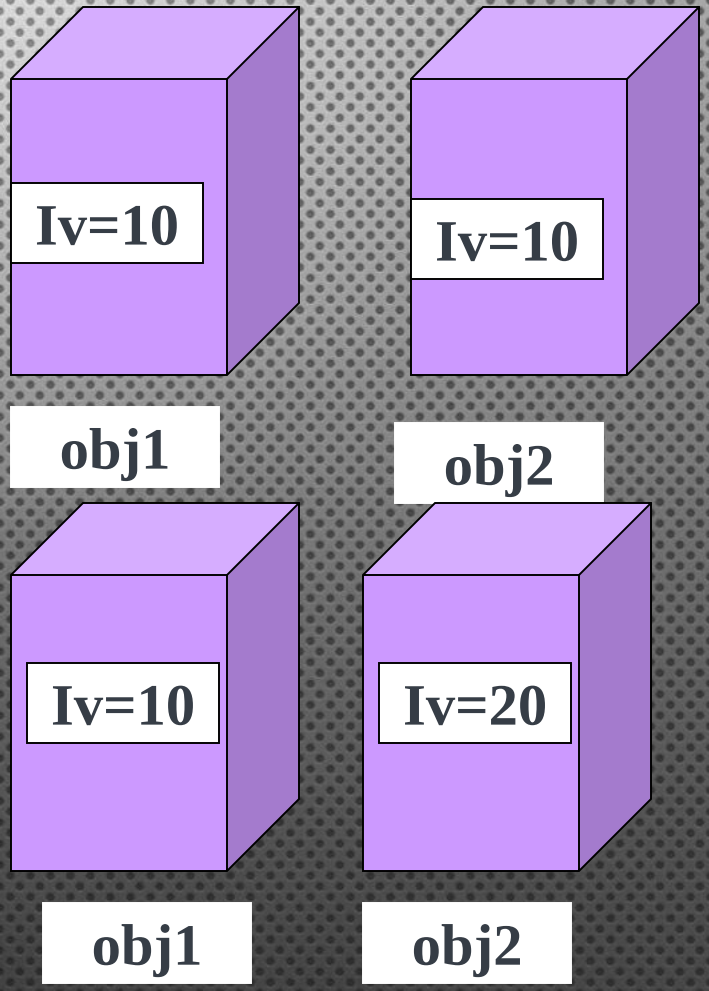
- These are collectively related to a class and none of the object can claim them its sole-proprietor .
- The variables defined with static keyword are shared by all objects.
- Here Objects do not store an individual value but they are forced to share it among themselves.
- These variables are declared as "static fields" using the static keyword.
- These variables are like global variables which are same for all objects of the class. These variables take default values if not initialized.

THREE TYPES OF VARIABLES

```
public class variabledemo1 {
int a=10;   // Instance Variable
static int b=20; // Class or Static Variable
public static void main(String args[])     {
int c=30;   // Local Variable
System.out.println("Local Variable : "+c);
variabledemo1 obj=new variabledemo1();
System.out.println("Instance Variable : "+obj.a);
System.out.println("Class or Static Variable : "+variabledemo1.b);
System.out.println("Class or Static Variable : "+obj.b);
System.out.println("Class or Static Variable : "+b); } }
```

INSTANCE VS CLASS VARIABLE

```
public class Demo {
int iv=10;
public static void main(String args[]){
Demo obj1=new Demo();
Demo obj2=new Demo();
System.out.println("Initialy");
System.out.println(obj1.iv+" "+obj2.iv);
obj2.iv=20;
System.out.println("Now");
System.out.println(obj1.iv+" "+obj2.iv);
} }
```



INSTANCE VS CLASS VARIABLE

```
PUBLIC CLASS DEMO {
STATIC INT SV=10;
PUBLIC STATIC VOID MAIN(STRING ARGS[]){
DEMO OBJ1=NEW DEMO();
DEMO OBJ2=NEW DEMO();
SYSTEM.OUT.PRINTLN("INITIALY");
SYSTEM.OUT.PRINTLN(OBJ1.SV+" "+OBJ2.SV);
OBJ2.SV=20;
SYSTEM.OUT.PRINTLN("NOW");
SYSTEM.OUT.PRINTLN(OBJ1.SV+" "+OBJ2.SV);
 } }
```