JAVA VARIABLES

Variables in Java

- **Variable** is name of *reserved area allocated in memory*. In other words, it is a *name of memory location*. It is a combination of "vary + able" that means its value can be changed.
- It is the basic unit of storage in a program.
- The value stored in a variable can be changed during program execution.

THE GENERAL RULES FOR CONSTRUCTING NAMES FOR VARIABLES

- Names can contain letters, digits, underscores, and dollar signs
- Names should begin with a letter
- Names can also begin with $, underscore(keyword after 1.8 java version)
- Names are case sensitive ("myVar" and "myvar" are different variables)
- Names should not contain whitespace
- Reserved words (like Java keywords, such as int or String) cannot be used as names

JAVA IDENTIFIERS

I. All Java **variables** must be **identified** with **unique names**

II. These unique names are called **identifiers**

III. Identifiers can be short names (like x and y) or more descriptive names (age, sum, total Volume)

JAVA VARIABLES - EXAMPLES

Examples

- Valid Names
  - MyVariable
  - myvariable
  - MYVARIABLE
  - x
  - i
  - _myvariable
  - $myvariable
  - _9pins
  - andros

- Invalid Names
  - My Variable // Contains a space
  - 9pins // Begins with a digit
  - a+c // The plus sign is not an alphanumeric character
  - testing1-2-3 // The hyphen is not an alphanumeric character
  - O'Reilly // Apostrophe is not an alphanumeric character
  - OReilly_&_Associates // ampersand is not an alphanumeric character

JAVA VARIABLES - TYPES

Types of Variables

1) **Local Variables**

A variable which is declared inside the method is called local variable.

2) **Instance Variables**

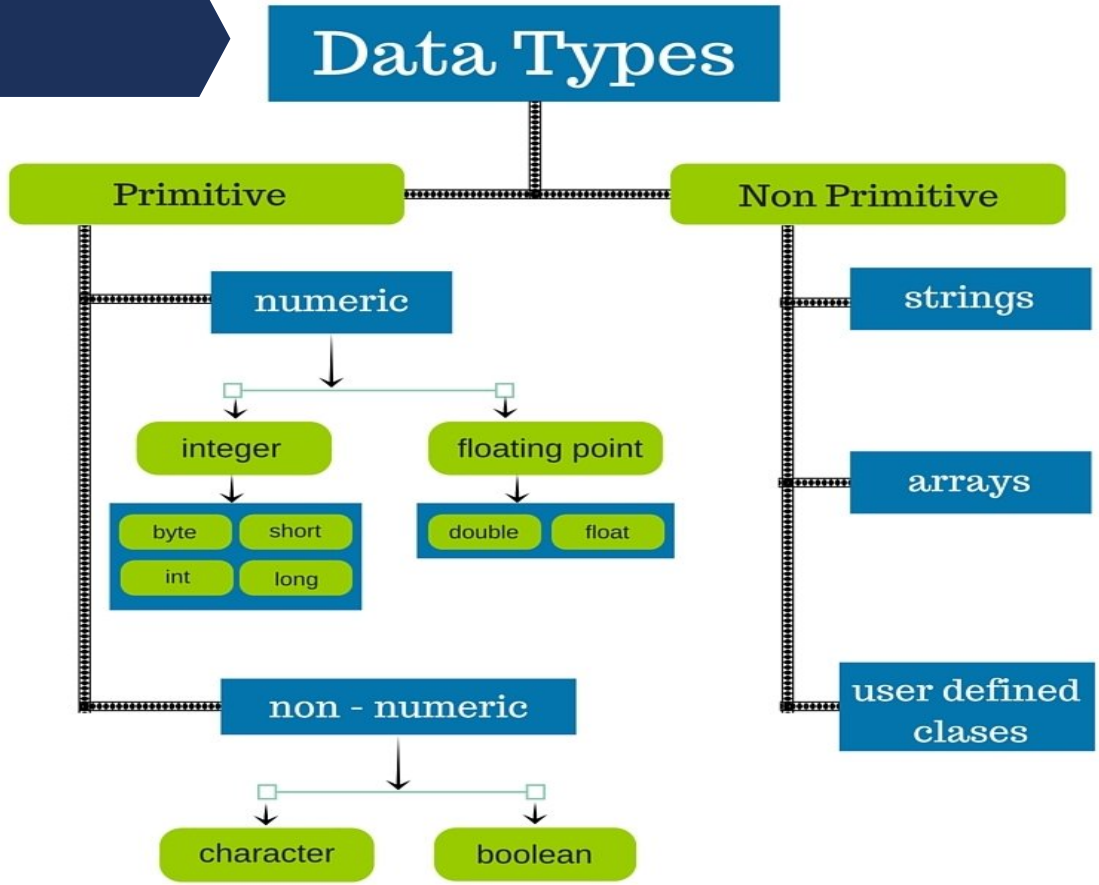
A variable which is declared inside the class but outside the method, is called instance variable . It is not declared as static.

3) **Static Variables**

A variable that is declared as static is called static variable. It cannot be local.

```
class A
{
  int data=50;//instance variable
  static int m=100;//static variable
  public static void main(String args[])
  {
   int n=90;//local variable
  }
}//end of class
```

DATA TYPES


Data Types
Primitive
Non Primitive
numeric
integer
floating point
byte
short
int
long
double
float
non - numeric
character
boolean
strings
arrays
user defined clases

DATA TYPES - SIZE

Sizes of DataTypes

Data Type	Default Value	Default Size
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

DECLARING (CREATING) VARIABLES

- To create a variable, you must specify the type and assign it a value

Syntax	**Example: 01**
type variable = value;	String name = "John"; System.out.println(name);

```
public class MyClass {
    public static void main(String[] args) {
        String name = "John";
        System.out.println(name);
    }
}
```

DECLARING (CREATING) VARIABLES

Syntax	Example: 02

int myNum = 5;
float myFloatNum = 5.99 f;
char myLetter = 'D';
boolean myBool = true;
String myText = "Hello";
	public class MyClass {
	  public static void main(String[] args) {
	  int myNum = 15;
	  System.out.println(myNum);
	  }
	}

VARIABLES - DISPLAY

The println() method is often used to display variables

- To combine both text and a variable, use the + character

Example: 04

String name = "John";
System.out.println("Hello " + name);
	public class MyClass {
	public static void main(String[] args) {
	String name = "John";
	System.out.println("Hello " + name);
	}
	}

LITERALS

- Value to be assigned for variable.
- Three types of literals:

  Primitive type

  String literals

  Null literals

JAVA TYPECASTING

- You really don’t want to spill that...

- Be sure the value can fit into the variable.

Example

```
int x = 24;
byte b = x;
//won’t work!!
```

JAVA TYPECASTING

Type casting is when you assign a value of one primitive data type to another type

- In Java, there are two types of casting
  - **Widening Casting (automatically) -** converting a smaller type to a larger type size

    byte -> short -> char -> int -> long -> float -> double

  - **Narrowing Casting (manually) -** converting a larger type to a smaller size type
  - double -> float -> long -> int -> char -> short -> byte

WIDENING CASTING

- Widening casting is done automatically when passing a smaller size type to a larger size type

Example

```
public class Main{
    public static void main(String[] args) {
        int myInt = 9;
        double myDouble = myInt;
        System.out.println(myDouble);
        System.out.println(myInt);
    }
}
```

NARROWING CASTING

- Narrowing casting is done automatically when passing a larger size type to a smaller size type

Example

```
public class Main{
    public static void main(String[] args) {
        double myDouble = 9.78;
        int myInt = (int)myDouble;
        System.out.println(myDouble);
        System.out.println(myInt);
    }
}
```

GUESS THE OUTPUT

```
public class Test {
    public static void main(String[] argv){
        char ch = 'c';
        int num = 88;
        ch = num;
    }
}
```

Answer: **Error**

GUESS THE OUTPUT

```
class Simple {
    public static void main(String[] args) {
        float f = 10.5 f;
        int a=(int)f;
        System.out.println(f);
        System.out.println(a);
    }
}
```

A. 10.5
10

B. 10
10

C. Error

GUESS THE OUTPUT

```
public class Test {
    static void test(float x) {
        System.out.print("float");
    }
    static void test(double x) {
        System.out.print("double");
    }
    public static void main(String[] args) {
        test(99.9);
    }
}
```

A. float

B. double ✓

C. Compilation Error

D. Exception is thrown at runtime

GUESS THE OUTPUT

```
public class Test {
    public static void main(String[] args) {
        int i = 010;
        int j = 07;
        System.out.println(i);
        System.out.println(j);
    }
}
```

A. 8 7 ✓

B. 10 7

C. Compilation fails with an error at line 3

D. Compilation fails with an error at line 5

E. None of these

GUESS THE OUTPUT

Of the below, what is an invalid variable name?

A. Shiva _ Mani _ 98480 _ 22338

B. 1942 _ a _ love _ story

C. Balu _ ABCDEFG

D. s _ ss _ ssshh

E. None of the above

Answer : B