Software Academy

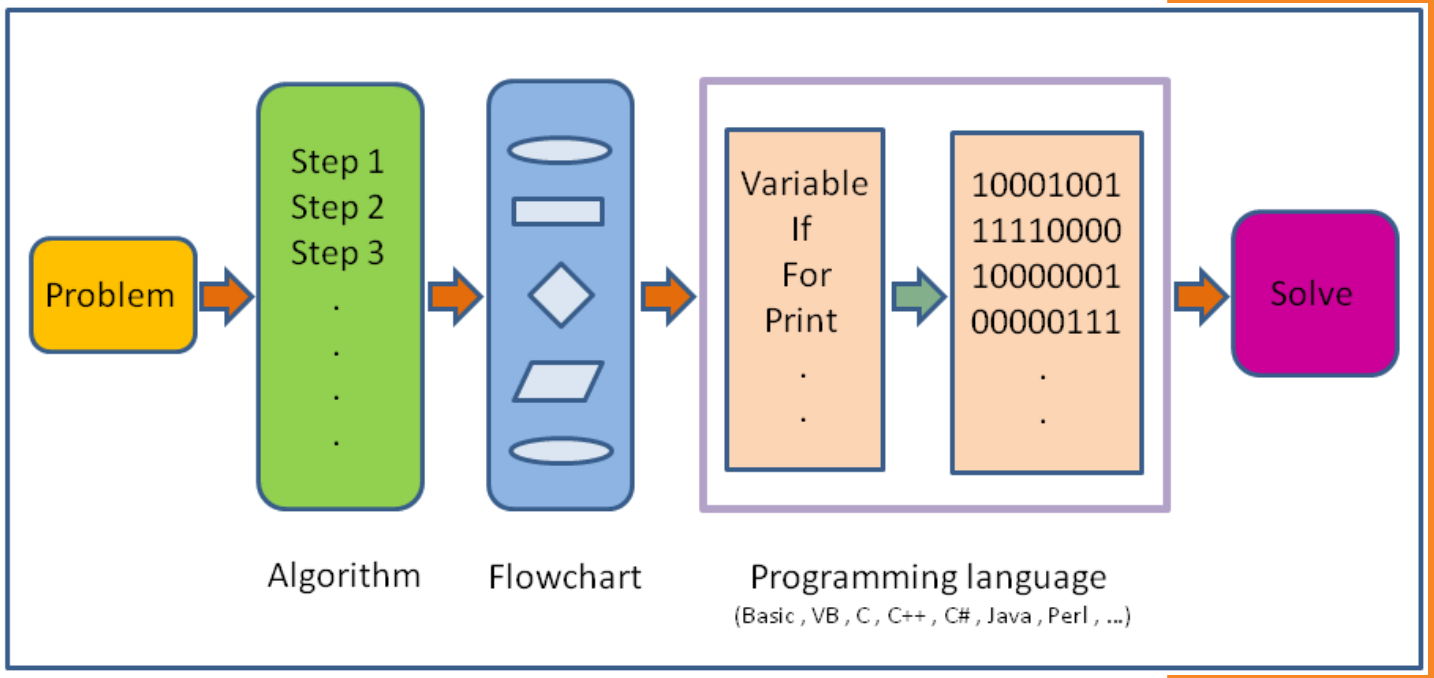
Back-end development: Java

TECH TALENT ACADEMY

Session Objective

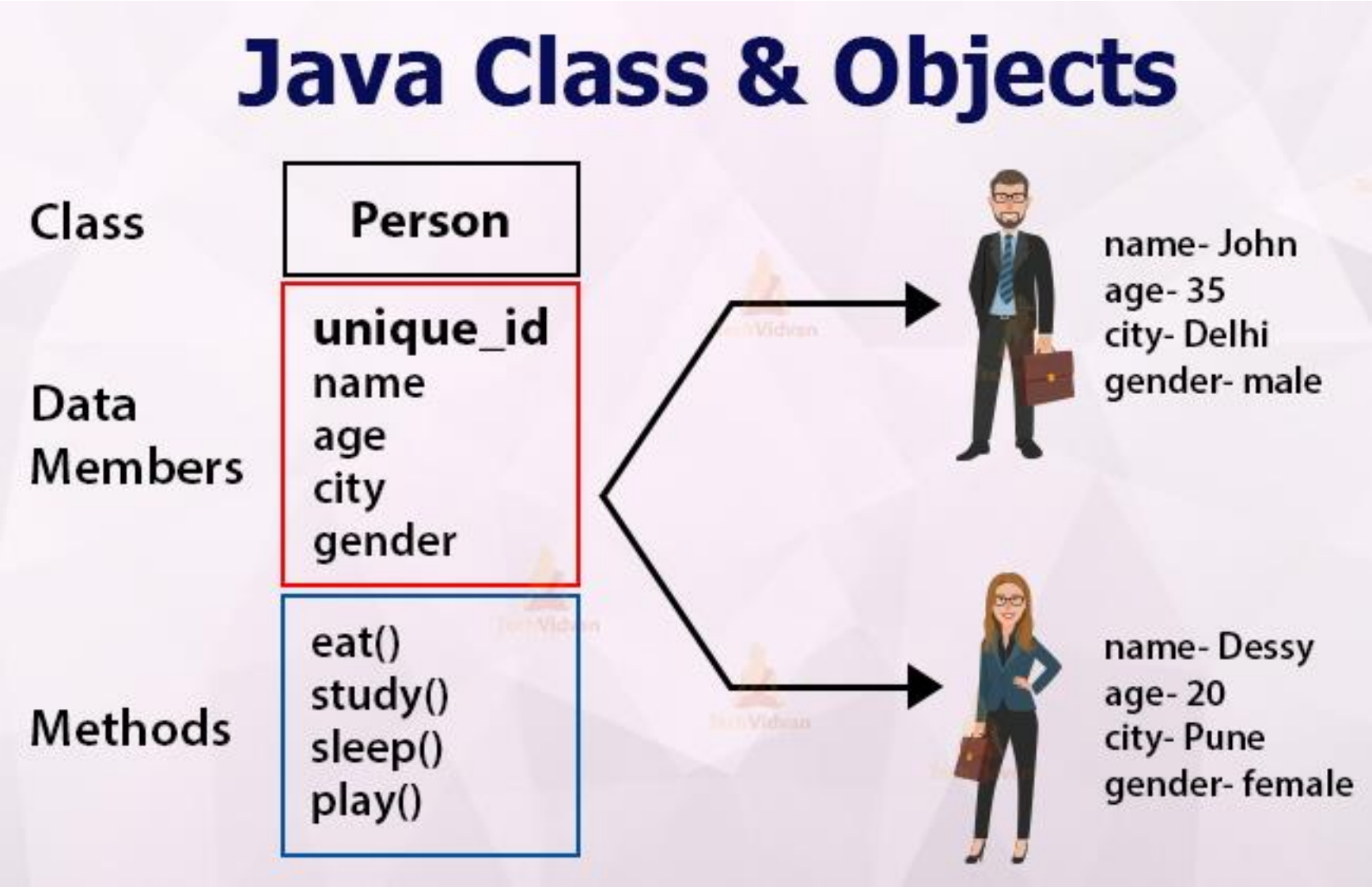
TTA

What is Programming Language?



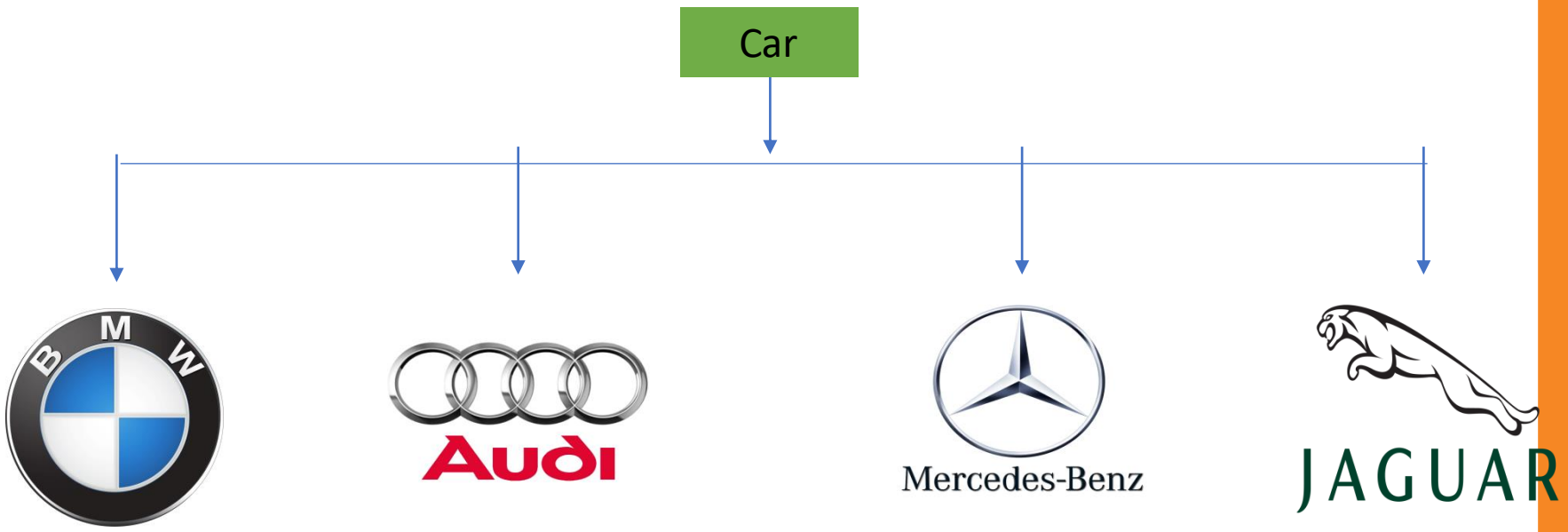
What is Java

- Java is an object oriented programming language.
- Java is created by “Sun Microsystems”, invented by James Gosling in 1991.
- Before ,Java was known as “Oak”.
- In the beginning java programming language was using for developing software for electronics devices.


Java Class & Objects
Class
Person
Data Members
unique_id
name
age
city
gender
Methods
eat()
study()
sleep()
play()
name- John
age- 35
city- Delhi
gender- male
name- Dessy
age- 20
city- Pune
gender- female

TTA

Defining a Class:



This is Car class ,all the cars have some common attributes like ,

1) All cars have engine
2) Cars have four wheels
3) Car have steering

So all the Brand cars belongs to Car class

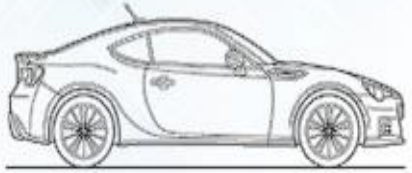
Identifying the Building Blocks of a Java Program(Contd..)

What is a Class?

Classes

- A class in java is a blueprint which includes all the data.
- It describes the state and behavior of a specific object.

Example :



Syntax :

```
Public class Car {

Color
Model
Price
speedUp();
gearChange();
}
```

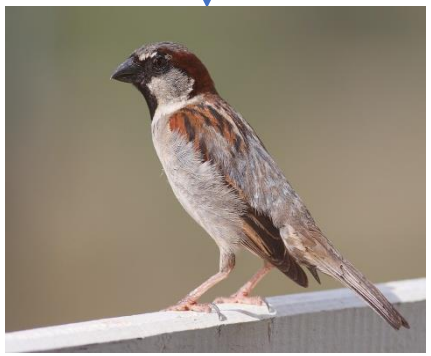
Class name

Variables

Methods

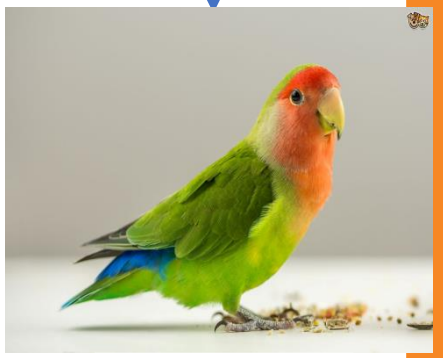
Defining a Class:

Birds



Sparrow

Crow



Parrot

Sparrow , Crow , Parrot these all are birds therefore they belongs to class Bird

1. Birds can fly
2. Birds have wings
3. Birds have Beak

TTA

Identifying the Building Blocks of a Java Program(Contd..)

```
class Birds
{
    int number_of_legs ;
    String bird_color ;

    public void fly() {

    }
    public void eat(){

    }
}
```

variables

Methods

Identifying the Building Blocks of a Java Program(Contd..)

- **A class defines:**
  - **The Properties and behavior of an object.**
  - **class is a data-type created by the user for own purpose.**
- The following code snippet shows how to declare a class:
- **Syntax:**

```
class <ClassName>
{
Declaration of member variables (properties)
Declaration of  member methods (behavior)
}
```

```
Example :-
class Student
{
 int  sid ; // Variable
 public void getData(){} // Method
}
```

Identifying the Building Blocks of a Java Program(Contd..)

- There are certain — **rules** that should be followed to **name Java classes**.

- Some of these rules are:
  - The name of a class should not contain any embedded space or symbol, such as ?, !, #, @, %, &, {}, [], :, ;, “, and /.
  - A class name must be unique .
  - **A class name must begin with a letter, an underscore (_), or the dollar symbol ($).** Or,
  - it must begin with an alphabet that can be followed by a sequence of letters or digits (0 to 9), ‘$’, or underscore(“ _ “) .
  - A class name should **not consist of a keyword.**

Identifying the Building Blocks of a Java Program(Contd..)

Int a_bc = 10; ✅

Int _abc = 5; // ✅

Int 2abc = 10 ; // cant start with digit ❌

Int abc2= 10 ; ✅

Int ab@c = 10 ; cant use any special symbol except “ _ “ ❌

Int if = 3; // keyword cant be used as variable name ❌

Int a bc = 20; // space is not allowed in variable name ❌

Identifying the Building Blocks of a Java Program(Contd..)

- **Keywords:**
  - Are the **reserved words with a special meaning for a language**, which **express the language features**

Identifying the Building Blocks of a Java Program(Contd..)

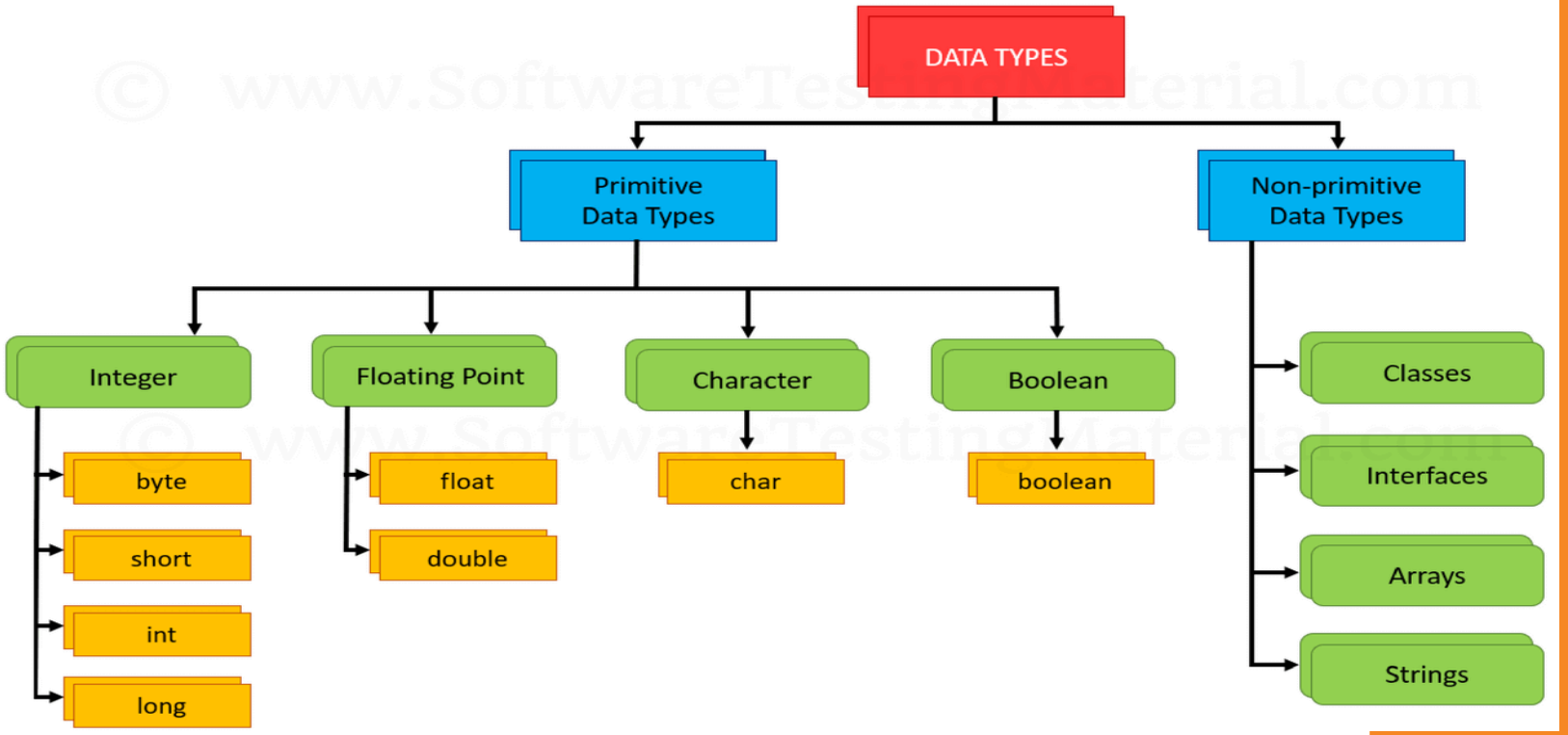
The following table lists the Java keywords.

abstract	boolean	break	byte
case	catch	char	class
const	continue	default	do
double	else	extends	final
finally	float	for	goto
if	implements	import	instanceof
int	interface	long	native
new	package	private	protected
public	return	short	static
strictfp	super	switch	synchronized
this	throw	throws	transient
try	void	volatile	while
enum	assert		

- The following naming conventions should be followed for naming a Java file:
  - A file name must be unique .
  - A file name cannot be a keyword.
  - If a class is specified as **`public`**, the file name and class name should be the same.
  - If a file contains multiple classes, only one class can be declared as `public`.
  - The file name should be the same as the class name that is declared `public` in the file.
  - If a file contains multiple classes that are not declared `public`, any file name can be specified for the file.

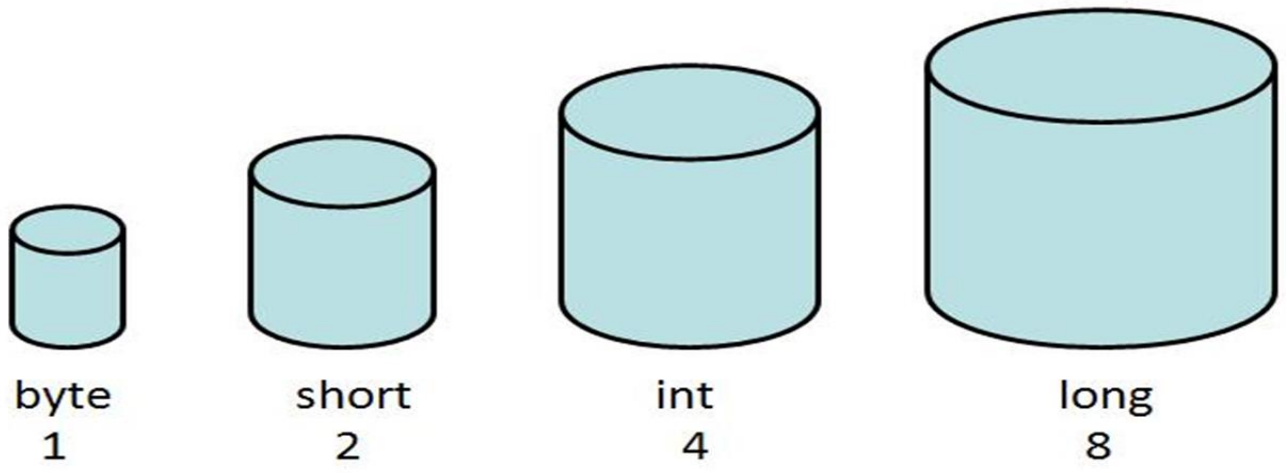
Identifying Data Types

- It tells what type of values can be stored.
- It defines the operations that can be done on the data.
- It defines a range of values that can be stored
- Defines the memory size for a variable.



Identifying the Building Blocks of a Java Program(Contd..)

Integer data types:



Identifying the Building Blocks of a Java Program(Contd..)

Primitive Data Types

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

Why char uses 2 byte in java and what is \u0000 ?
because java uses unicode system rather than ASCII code system. \u0000 is the lowest range of unicode system.

TYPE	DESCRIPTION	DEFAULT	SIZE	EXAMPLE LITERALS	RANGE OF VALUES
boolean	true or false	false	1 bit	true, false	true, false
byte	twos complement integer	0	8 bits	(none)	-128 to 127
char	unicode character	\u0000	16 bits	'a', '\u0041', '\101', '\\', '\','\n',' β'	character representation of ASCII values 0 to 255
short	twos complement integer	0	16 bits	(none)	-32,768 to 32,767
int	twos complement integer	0	32 bits	-2, -1, 0, 1, 2	-2,147,483,648 to 2,147,483,647
long	twos complement integer	0	64 bits	-2L, -1L, 0L, 1L, 2L	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	IEEE 754 floating point	0.0	32 bits	1.23e100f, -1.23e-100f, .3f, 3.14F	upto 7 decimal digits
double	IEEE 754 floating point	0.0	64 bits	1.23456e300d, -1.23456e-300d, 1e1d	upto 16 decimal digits

Identifying the Building Blocks of a Java Program(Contd..)

```
// Java program to explain primitive data types in Java

    class Test {
  public static void main(String args[])  {

    char a = ‘G‘; // declaring character
    int i = 89; // Integer data type is generally used for numeric values

    byte b = 4; // use byte and short if memory is a less

   / / byte b1 = 7888888955; // this will give error as number is  larger than byte range

    short s = 56;

   // short s1 = 87878787878; // this will give error as number is larger than short range

    double d = 4.355453532;  //  Error by default fraction value is double in java

    float f = 4.7333434f;   // for float use 'f' as suffix

    System.out.println(“char: ” + a);   System.out.println("integer: " + i);       System.out.println("byte: " + b);
    System.out.println("short: " + s);
    System.out.println("float: " + f);
    System.out.println("double: " + d);
  }
}
```

TTA

Wrapper class:

- In order to use the primitive data types as objects, Java provides wrapper classes .
- A wrapper class acts like an object wrapper.
- The various wrapper classes, provided by the `java.lang` package. They are:

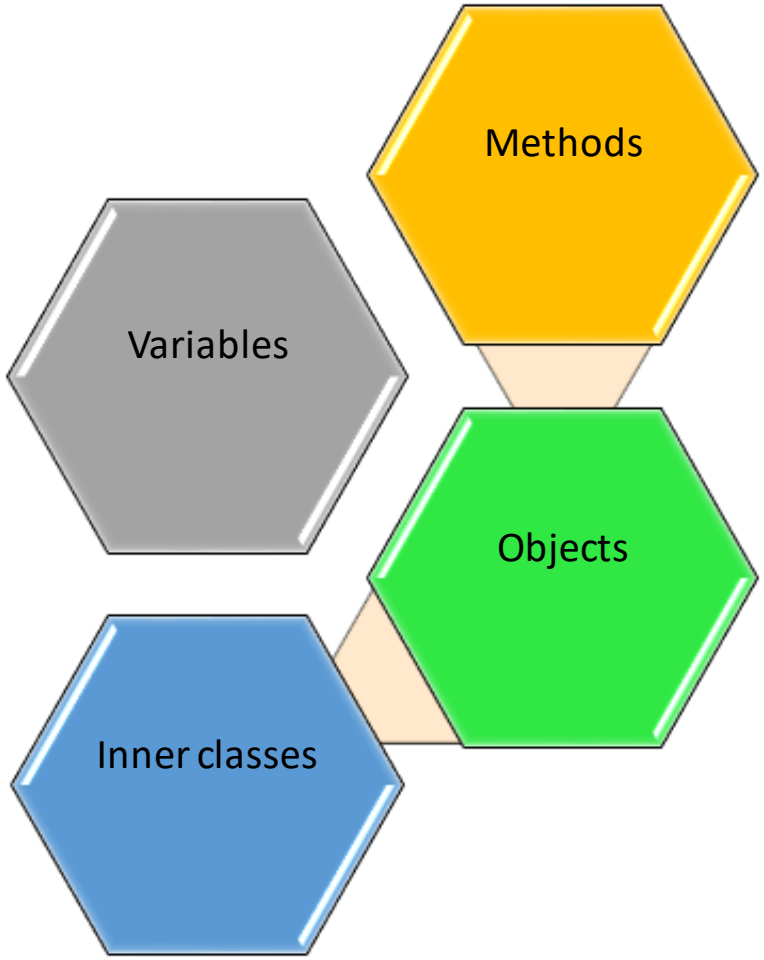
Primitive Data Type	Wrapper Class
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean

Identifying the Building Blocks of a Java Program(Contd..)

In order to get minimum and maximum value we use **Wrapper classes MAX_VALUE and MIN_VALUE constant.**

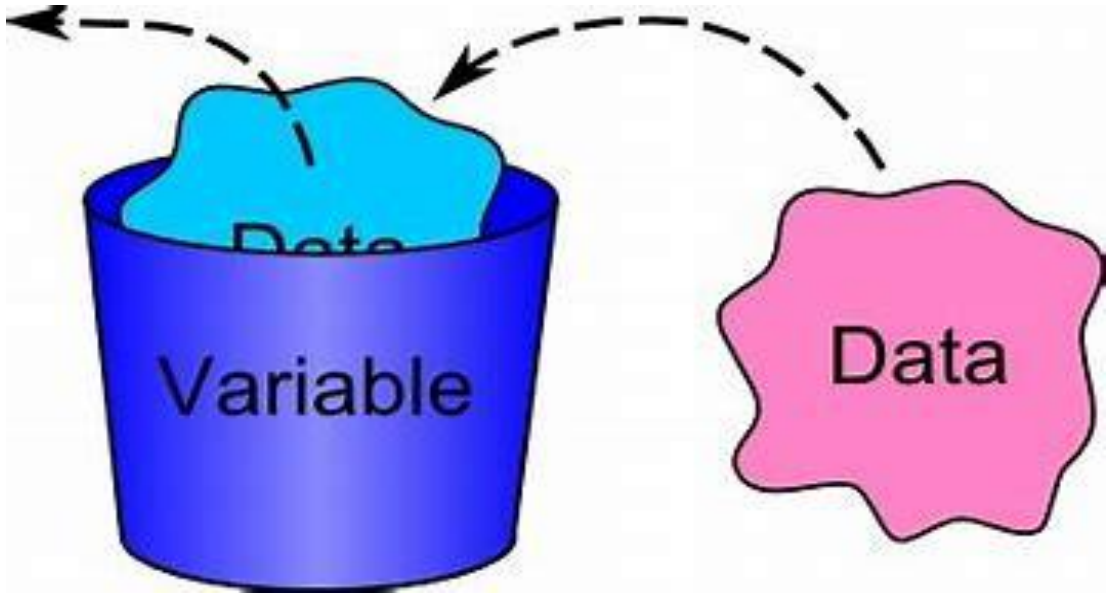
```
//Program to demonstrate how get and use minimum and maximum values of primitive data types in Java.

class MinAndMaxDemo
{
 public static void main(String[] args)
 {
  Byte minimumByteValue = Byte.MIN_VALUE;
  Byte maximumByteValue = Byte.MAX_VALUE;
  System.out.println("Minimum byte value is : " + minimumByteValue);
  System.out.println("Maximum byte value is : " + maximumByteValue);

  Short minimumShortValue = Short.MIN_VALUE;
  Short maximumShortValue = Short.MAX_VALUE;
  System.out.println("Minimum short value is : " + minimumShortValue);
  System.out.println("Maximum short value is : " + maximumShortValue);
```

Identifying Class Members



Identifying the Building Blocks of a Java Program(Contd..)

What is Variable

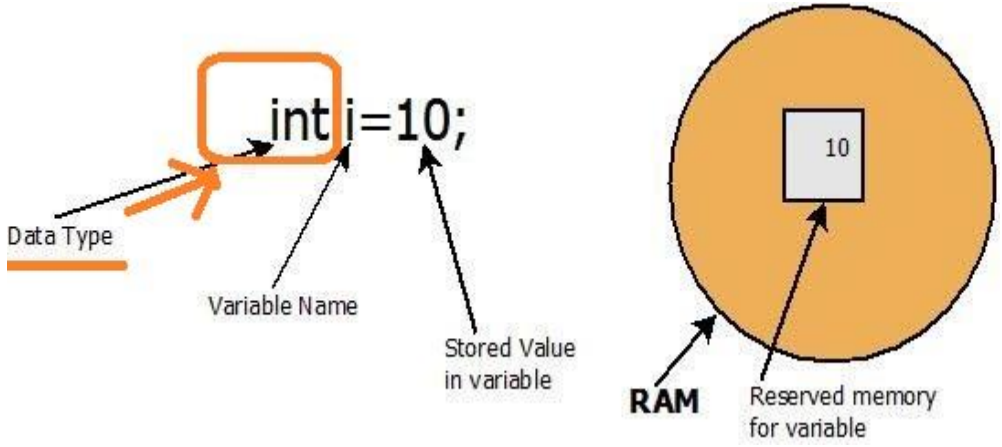


What is Variable

- A variable:
  - used essentially as a container for the storage of varying kinds of data.
  - Represents a name that refers to a memory location where some value is stored.
  - **Needs to be declared before it is accessed.**

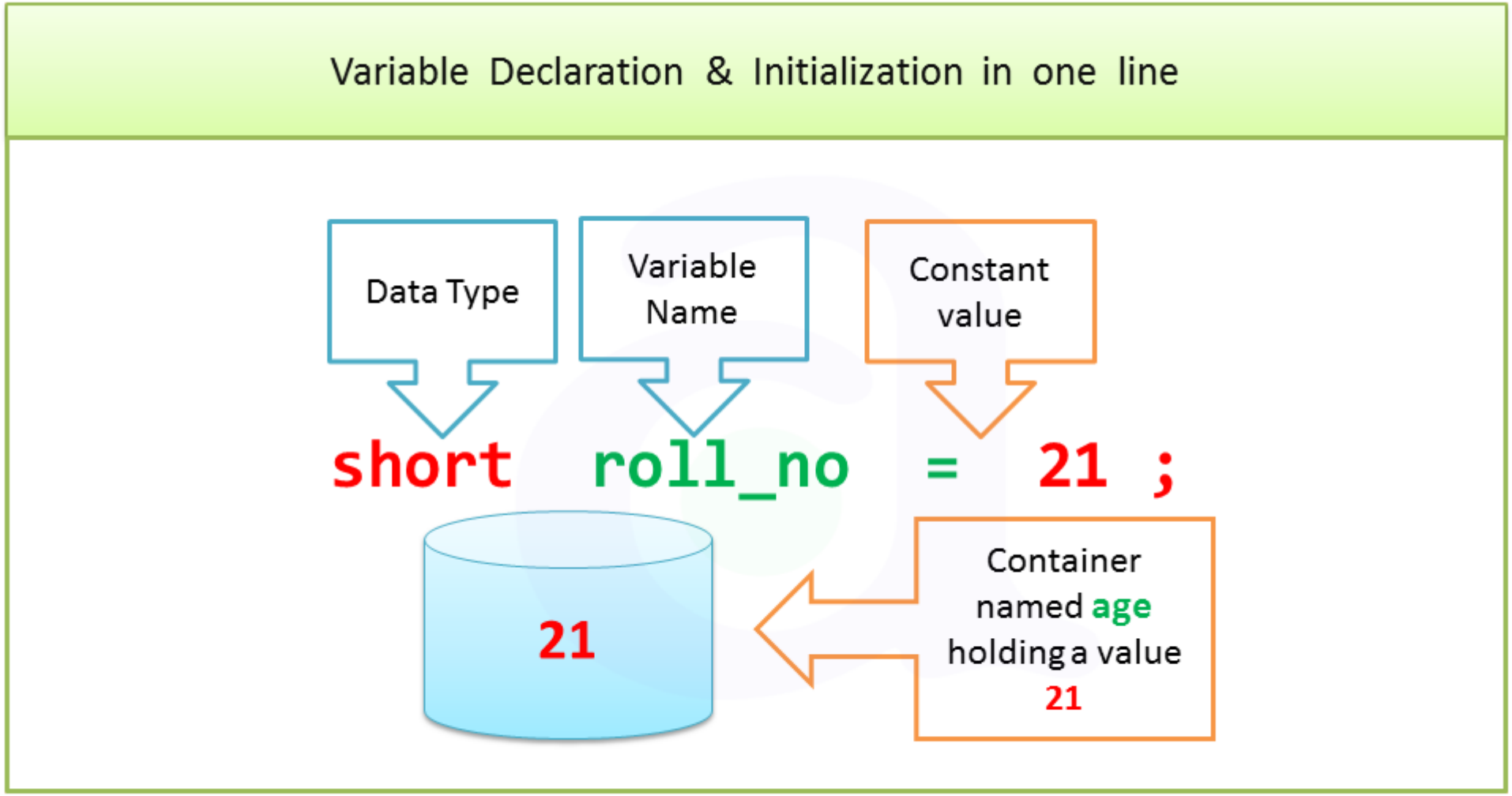
Identifying the Building Blocks of a Java Program(Contd..)

- The following code snippet shows how to declare a variable




```
Example : To store the integer value for ID of students
int StudentID;
```

Identifying the Building Blocks of a Java Program(Contd..)



short variable Declaration and Initialization

Identifying the Building Blocks of a Java Program(Contd..)

Instance Variables (variables inside class):

The variables defined inside the class but outside the method is called Instance variables.

Local Variables:

The variables defined inside the method or constructor is called Local variables.

Static Variables:-

The variable defined inside class using **"static" keyword** ,but outside the methods are called static variables.

Identifying the Building Blocks of a Java Program(Contd..)

```
class  Student
{
int sid ;
String sname;

static String university = "Ningxia";

void setData(int id , String name)
{
sid = id ;
sname = name;
}
}
```

instance variables

static variable

local variable

Identifying the Building Blocks of a Java Program(Contd..)

- Widening Casting(Implicit)

Done by complier for us Automatically

- Narrowing Casting(Explicitly done)

We need to request to complier using “type-casting”

What is Literal

- A literal:
  - A value that is assigned to a variable or constants .
  - Contains a sequence of characters , such as digits , alphabets , or any other symbol , which represents the value to be stored.

Identifying the Building Blocks of a Java Program(Contd..)

- The various types of literals in Java are:

Integer literals
- Are non-fractional numeric values.
- **Int a = 5;**

Floating point literals
- Are numeric values that contain a fractional part.
- **Float x = 4.5f , double x = 4.2;**

Boolean
- Can store only the values, `true` and `false`.
- Boolean a= true ,
- b= false

Character type
- Can store a single character, such as a symbol, letter, and number.
- **Char ch = 'A' ;**

String literals
- Are enclosed in double quotation marks.
- **String s = "Hello" ;**

Binary literals
- Are the literals that can be used to express the primitive integer data types into binary form.
- **Int a = 0b101;**

Identifying the Building Blocks of a Java Program(Contd..)

```
// Java program to show how to use literals in java
 class Test {
   public static void main(String[] args)
   {
      int a = 5; // integer literal
      float b = 4.5 f; // float literal
      double c = 4.5; // double literal
       boolean result = true ; // boolean literal
      char d = 'A' ; // character literal
       String name = "Peter"; // String literal
      System.out.println(a);
      System.out.println(b);
      System.out.println(c);
      System.out.println(result);
      System.out.println(d);
      System.out.println(name);
   }
}
```

Identifying the Building Blocks of a Java Program(Contd..)

For Integral data types (byte, short, int, long), we can specify literals in 4 ways:-

1. **Decimal literals (Base 10)** : In this form the **allowed digits are 0-9.**

int x = 25;

2. **Octal literals (Base 8)** : In this form the **allowed digits are 0-7**, octal number should be **prefix with 0.**

int x = 0146;

3. **Hexa-decimal literals (Base 16)** : In this form the **allowed digits are 0-9 and characters are A to F(a-f).** We can use both uppercase and lowercase characters. The hexa-decimal number should be **prefix with 0X or 0x.**

int x = 0X123Face;

4. **Binary literals** : we can specify literals in binary form also, **allowed digits are 0 and 1**. Literals value should be **prefixed with 0b or 0B.**

int x = 0b1111;

Identifying the Building Blocks of a Java Program(Contd..)

```
// Java program to illustrate the application of Integer literals
 class Test {
   public static void main(String[] args)
   {
      int a = 101; // decimal-form literal
      int b = 0100; // octal-form literal
      int c = 0xFace; // Hexa-decimal form literal
      int d = 0b1111; // Binary literal
      System.out.println(a);
      System.out.println(b);
      System.out.println(c);
      System.out.println(d);
   }
}
```

Identifying the Building Blocks of a Java Program(Contd..)

Important points for “Literals” in java

1. Every Integer literal by default type of “int

For example:-

```
byte x = 5 ; // ok
x = x+1; // error
x++ ; // ok
```

Same is true for short

```
long x = 5; // 5 is “int” so for long we use “L/l”

long x = 5L ; // correct for long
```

2. Every floating point literal is “double” by default.

```
float x = 4.5 ; // error (Narrowing of data type is not allowed in java)
float x = 4.5f ; // ok
```

- A method:
  - Method is a group of instructions to do some task.
  - Consists of two parts, method declaration and method body.
  - **Syntax:**

```
<Access specifier> <Return type> <Method name>(Parameter list)
{
    body of the method
}
```

- Access specifier: Public, Private, Default(No Keyword, Protected
- Return type: void(No return), Data type(int, float, char , String etc.
- Method name: Any name like variable-name
- Parameter list: Any data-type, any number, any order if no parameter empty ()

Identifying the Building Blocks of a Java Program(Contd..)

- A method:

```
public void add(int a, int b)
{
    c=a + b;
    System.out.println(c);
}
```

Return type

Parameter list

Method Name

Access Specifier

Method body

Identifying the Building Blocks of a Java Program(Contd..)

Access Modifer

return data type

method name

parameters passing to method

```
public int  calculateSUM(int a, int b)   {
    int c = a + b;
    return c;
}
```

statement

returning result

Rules for Methods:

- The first letter of the method should be in lowercase.
- If the method name consists of several words, the first letter of each word, except the first word, should be capitalized.
- **Single-word method name:** sum(), area()
- **Multi-word method name:** areaOfCircle(), stringComparision()

- The syntax for calling a method is:

```
<Method name>(argument list);
```

Identifying the Building Blocks of a Java Program(Contd..)

```
public class Test {
    public static void sayHello()
    {
        System.out.println("Hello");
    }
    public static void main(String[] args) {
        System.out.println("Hi im main method");
        sayHello();
    }
}
```

static means no need to create an object to call the method

Program execution starts from main() in java
在java中， 程序执行从main （）开始

Method calling

Output

```
System.out.println("Hi im main method");

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

Identifying the Building Blocks of a Java Program(Contd..)

```
public class Test {
   public static int max(int numA, int numB)
   {
     int result ;
      if(numA>numB)
      {
          result = numA;
     }
      else
      {
        result = numB ;
       }
        return result;
    }
   public static void main(String[] args) {
        int a = 15 , b = 8 ;
         int c = max(a, b);
      System.out.println("Largest value = "+ c);
   }
}
```

Identifying the Building Blocks of a Java Program(Contd..)

```java
public static void main(String args[])
{

int a = 15;
int b = 8;
int c = max(a, b);
System.out.println("Largest Value = " + c);

}
```

Called from main function

```java
public static int max(int numA, int numB) {
        int result;

        if (numA > numB){
                result = numA;
        }
        else{
                result = numB;
        }

        return result;
}
```

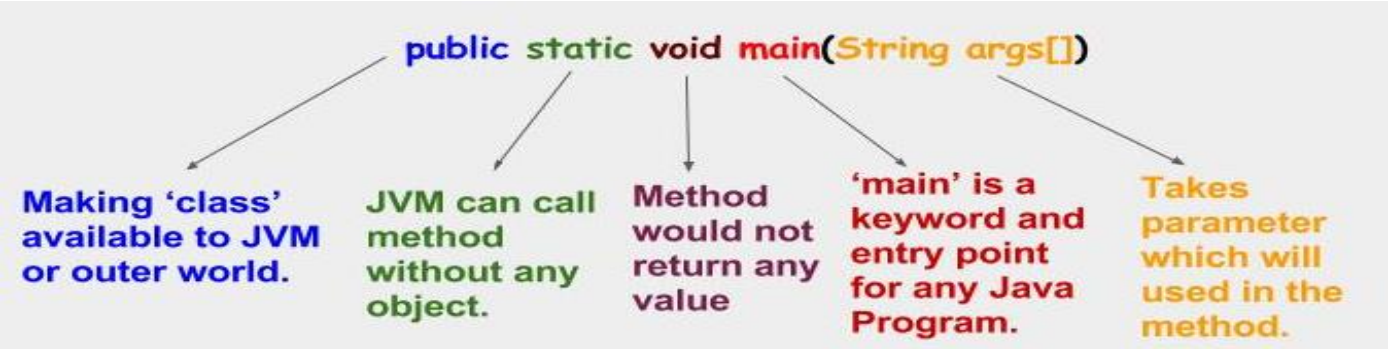
Calling function max

the flow of control transfers to calling function . Once the max method is finished, it returns control back to the caller.

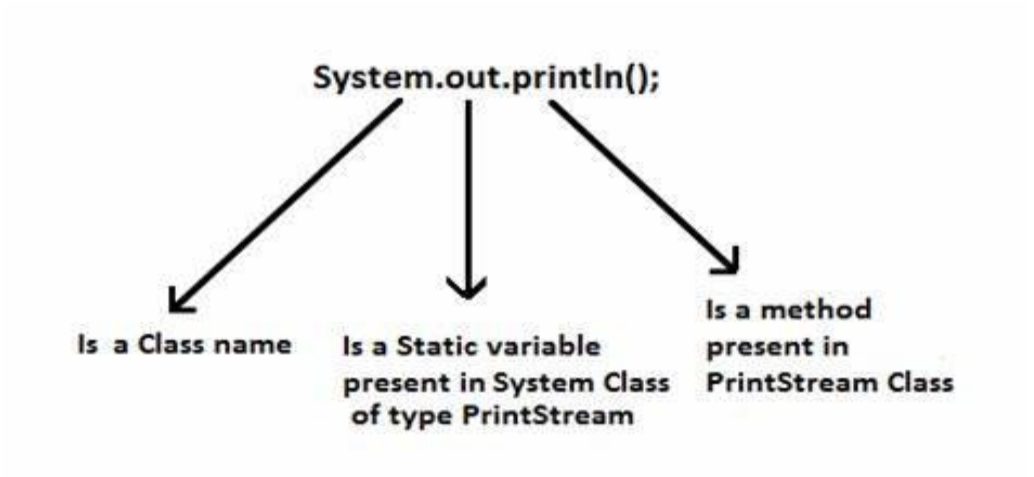
Identifying the Building Blocks of a Java Program(Contd..)

```
<Access Modifier> <Non-access modifier> <Return type> <Method Name> (Param_1,Param_2,..)
{
        <Method Body>

        return(return_value);
}
```

Main method in Java

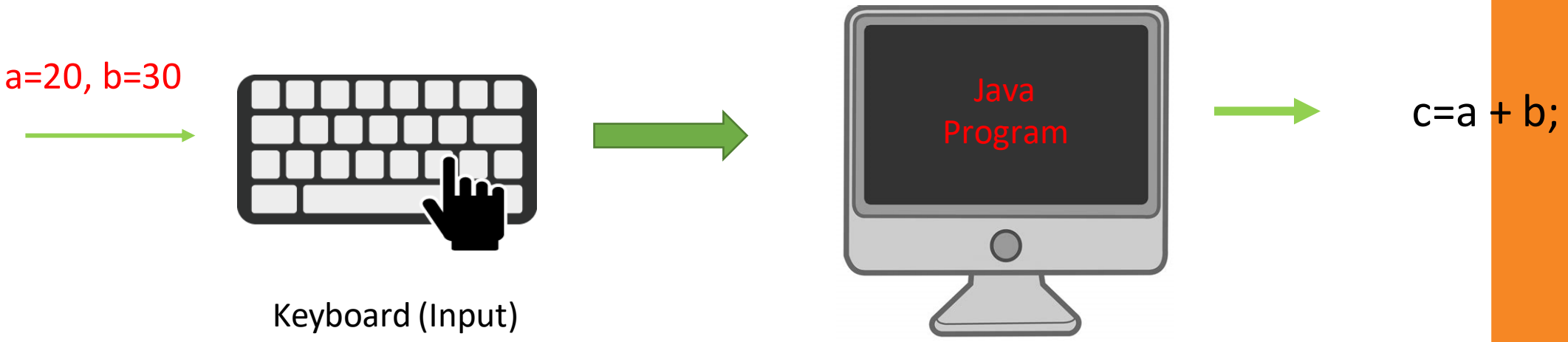


Identifying the Building Blocks of a Java Program(Contd..)



Identifying the Building Blocks of a Java Program(Contd..)

Scanner Class:



Scanner Class:

The Scanner Class

To read input from the keyboard we can use the `Scanner` class.

The `Scanner` class is defined in `java.util`, so we will use the following statement at the top of our programs:

```
import java.util.Scanner;
```

How to create object of Scanner class

```
Scanner sc = new Scanner(System.in);
```

Read the data from Keyboard

Provide the environment to read the data from keyboard

Scanner Methods

NAME	USE
nextInt();	Returns the next integer value
nextDouble();	Returns the next double value
nextFloat();	Returns the next float value
nextLong();	Returns the next long value
nextShort();	Returns the next short value
next();	Returns the next one word String value
nextLine();	Returns the next multiple word String value

Identifying the Building Blocks of a Java Program(Contd..)

Add 2 numbers using Scanner class

```
import java.util.Scanner
public class ScannerDemo
{
public static void main(String args[])
{

Scanner s=new Scanner(System.in);
System.out.println("Enter first no= ");

int num1, num2;

num1=s.nextInt();

System.out.println("Enter 2nd no:");

num2=s.nextInt();
System.out.println("Sum of no is= "+(num1+num2));
}
}
```

1. Import Scanner Class

2. Construct Scanner class Object

3. Define Variable to Receive Input

4. Read Input from Keyboard

TECH TALENT
ACADEMY