

OOP	Open JDK - Version 11.x	Java-Version
-----	----------------------------	--------------

JDK-> tools + docs + JRE -> tools + docs + rt.jar(core libraries) + JVM(Java Virtual Machine)	JRE + java tool
--	-----------------

javac	Program.java	
java	class Program{	javac -d ../bin Program.java
javap		-> Program.class
jar	public static void main(String [] args){	
	System.out.println("Hello world");	SET CLASSPATH=../bin
	}	java Program
	}	

Scanner , Console

Scanner sc = new Scanner(System.in);	Console con = System.console();
--------------------------------------	---------------------------------

Naming Convention

Camel Case	static and constant fields
- First letter of every word should be in capital except first word.	Everting in capital
- calculateTotalSalary	PI = 3.14;
- totalSalary	ROI = 5.6;
- name	static LOGIN_STATUS=true;
- will be used for (variables, fields, methods, method parameters)	

Pascal Case	packages
- First letter of every word should be in capital	the name of package should be in small case
- Employee	
- BankAccount	
- will be used for class, enum, Interface	

Keyword

- These are the reserved words in any language that have special mening to it.
- These are used to provide extra information to the compiler
- These cannot be used as identifiers.

Identifier	It has its own syntax
- It is a name given to the memory location	It has its own rules

Datatypes

- Datatype defines 3 things
- 1. Nature
 - What type of data can be stored inside them
- 2. Memory
 - The memory required to store the data.
- 3. Operation
 - Types of operations that can be carried on the data.

1. Primitive Datatype (Value types)

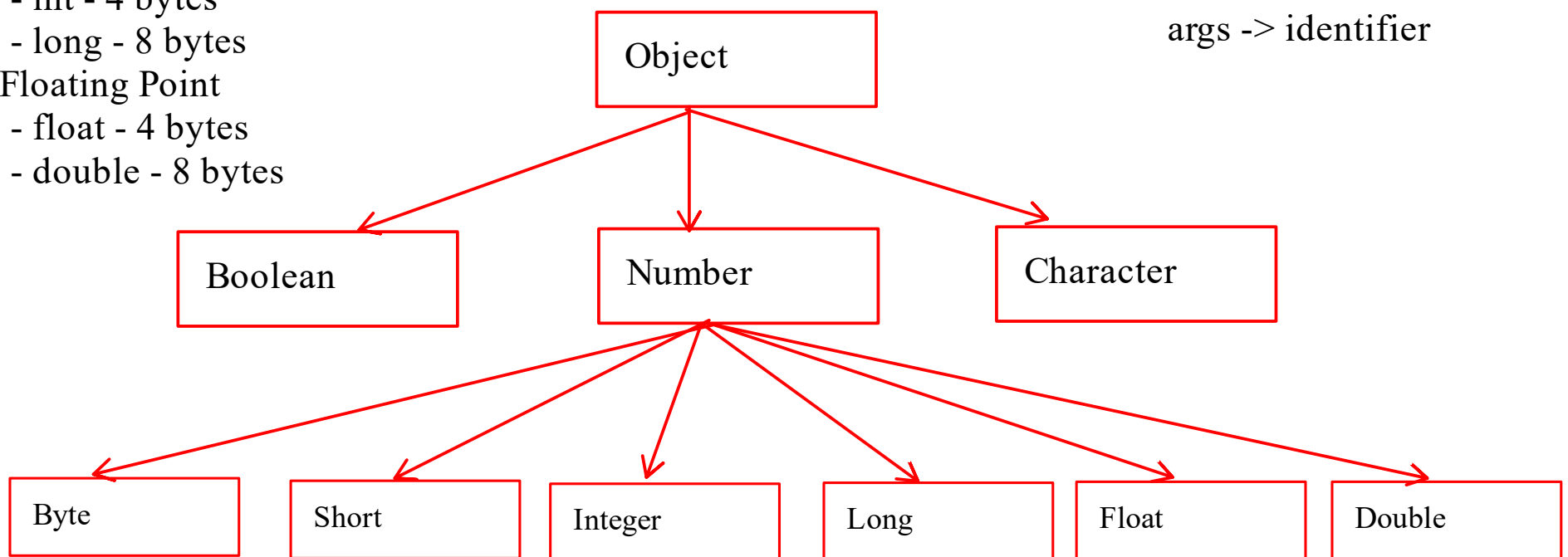
- a. Boolean
 - boolean - 1 bit
- b. Character
 - char - 2 bytes
- c. Integral
 - byte - 1 byte
 - short - 2 bytes
 - int - 4 bytes
 - long - 8 bytes
- d. Floating Point
 - float - 4 bytes
 - double - 8 bytes

2. Non-Primitive Datatype (Reference types)

- Class
- Array
- Enum
- Interface

main -> not a keyword

args -> identifier



Wrapper Class

- For every primitive type java have defined respective classes. These are called as Wrapper classes
- use of wrapper classes
 1. To get information about the datatype
 2. To access the helper methods from the classes
 3. Type conversion from primitive to reference type
 4. To store primitive type data in the collections

```
vector<Integer> v1;  
v1.push_back(10);
```

Widening

- Converting the narrower type of data into wider type is called as widening

Narrowing

- Converting the wider type of data into the narrower type is called as narrowing

```
class Time{  
int hrs;  
int mins;  
}  
  
main(){  
Time t1 = 120;  
Time t2(10,20);  
int minutes = t2;  
}
```

Boxing - Converting state of Primitive types to non primitive type

UnBoxing - Converting state of Non primitive type to Primitive type

- If the conversion is done automatically then it is called as Auto Boxing/UnBoxing
- conversion between the primitive type and their respective Wrapper class is always Auto Boxing and Auto UnBoxing
- i.e no need to call any methods for conversion.

java Program 10 20

Package

- It is a container

1. To Group replated types together
2. To organize the code

com.sunbeam.eattendance.entity

- Employee
- Student

com.sunbeam.eattendance.util

- DBUtil

com.sunbeam.eattendance.testers

- Program01
- Program02

java.lang

-

java.util

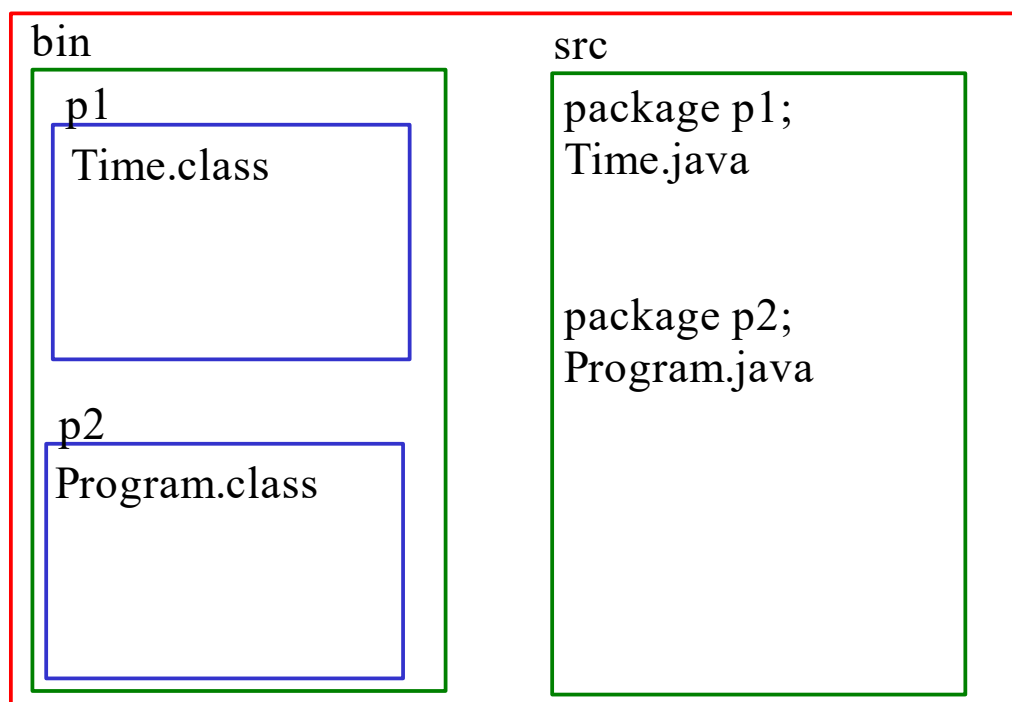
- Scanner
- Collection
- Date

java.io

java.sql

- Date

demo02



- Do all the compilation from src directory

```
javac -d ../bin Time.java
```

```
SET CLASSPATH=../bin
```

```
javac -d ../bin Program.java
```

```
java p2.Program
```

com.sunbeam.shape.dim2d

com.sunbeam.shape.dim3d

com.sunbeam.shape.testers

// Visibility modifiers for the members of the class

1. private

- visible only with the class and not outside the class

2. default

- If no visibility modifier is provided then it is considered as default.
- default has the visibility of package level private
- It can be visible within the same package and not outside the package.

3. protected

- it is visible inside the same package as well as in the sub class declared in other packages

4. public

- It is visible across all the packages

// visibility modifier for the class

1. default

- The class is visible only within that declared package and not outside the package

2. public

- The class is visible in the same package and also outside that declared package