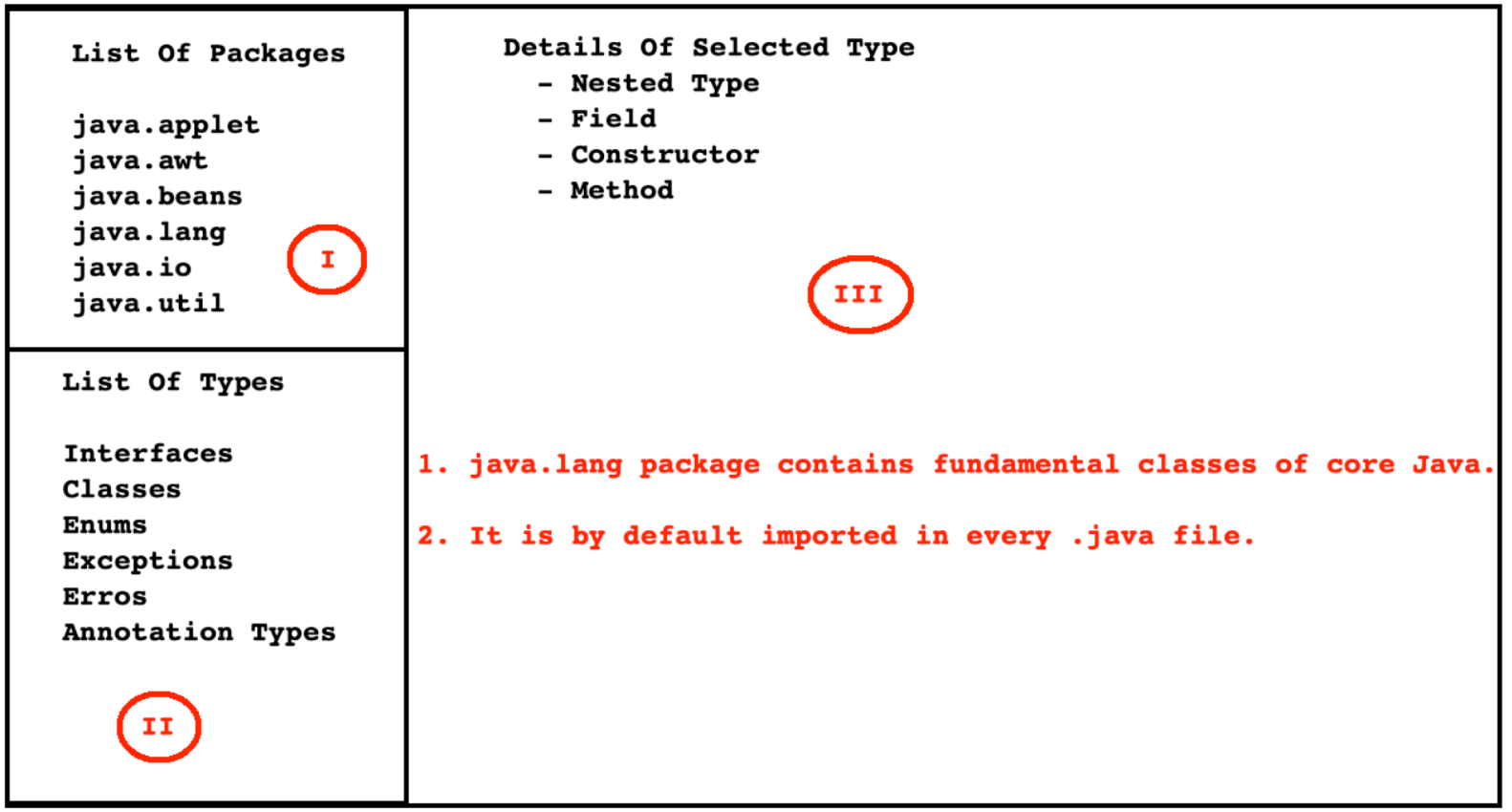
Java Standard Edition 8

Sandeep Kulange
sandeepkulange@sunbeaminfo.com

Java Docs


List Of Packages
java.applet
java.awt
java.beans
java.lang
java.io
java.util
I
List Of Types
Interfaces
Classes
Enums
Exceptions
Erros
Annotation Types
II
Details Of Selected Type
- Nested Type
- Field
- Constructor
- Method
III
1. java.lang package contains fundamental classes of core Java.
2. It is by default imported in every .java file.

java.lang.System class

```
package java.lang;
import java.io.*;
public final class System{
    public static final InputStream in;
    public static final OutputStream out;
    public static final OutputStream err;

    public static Console console();
    public static void exit(int status);
    public static void gc();
}
```

Stream

- Stream is an abstraction(object) which either produce(write)/consume(read) information from source to destination.
- Standard stream objects of Java which is associated with console:
    1. **System.in**
        - It represents keyboard.
    2. **System.out**
        - It represents Monitor.
    3. **System.err**
        - Error stream which represents Monitor.

How to access members of package?

Package : p1

```
public class Complex{
  //TODO
}
```

```
public class Program{
  public static void main( String[] args ){
    p1.Complex c1 = new p1.Complex( );
  }
}
```

1

```
import p1.Complex;
public class Program{
  public static void main( String[] args ){
    Complex c1 = new Complex( );
  }
}
```

2

User Input Using Console class.

- Console is class declared in java.io package.
- console( ) is a static method of System class which returns reference of java.io.Console class.
  - public static Console console( );
- **public String readLine( )** is a method of java.io.Console class.

```
java.io.Console console = System.console( );
String name = console.readLine( );
int empid = Integer.parseInt( console.readLine( ) );
float salary = Float.parseFloat( console.readLine( ) );
```

```
import java.io.Console;
Console console = System.console( );
String name = console.readLine( );
int empid = Integer.parseInt( console.readLine( ) );
float salary = Float.parseFloat( console.readLine( ) );
```

User Input Using Scanner class.

- Scanner is a final class declared in java.util package.
- Methods of Scanner class:
    1. public String nextLine()
    2. public int nextInt()
    3. public float nextFloat()
    4. public double nextDouble()

- How to user Scanner?

```
Scanner sc = new Scanner(System.in);
String name = sc.nextLine( );
int empid = sc.nextInt( );
float salary = sc.nextFloat( );
```

Modifier

1.ABSTRACT

2.FINAL

3.INTERFACE

4.NATIVE

5.PRIVATE

6.PROTECTED

7.PUBLIC

8.STATIC

9.STRICT

10.SYNCHRONIZED

11.TRANSIENT

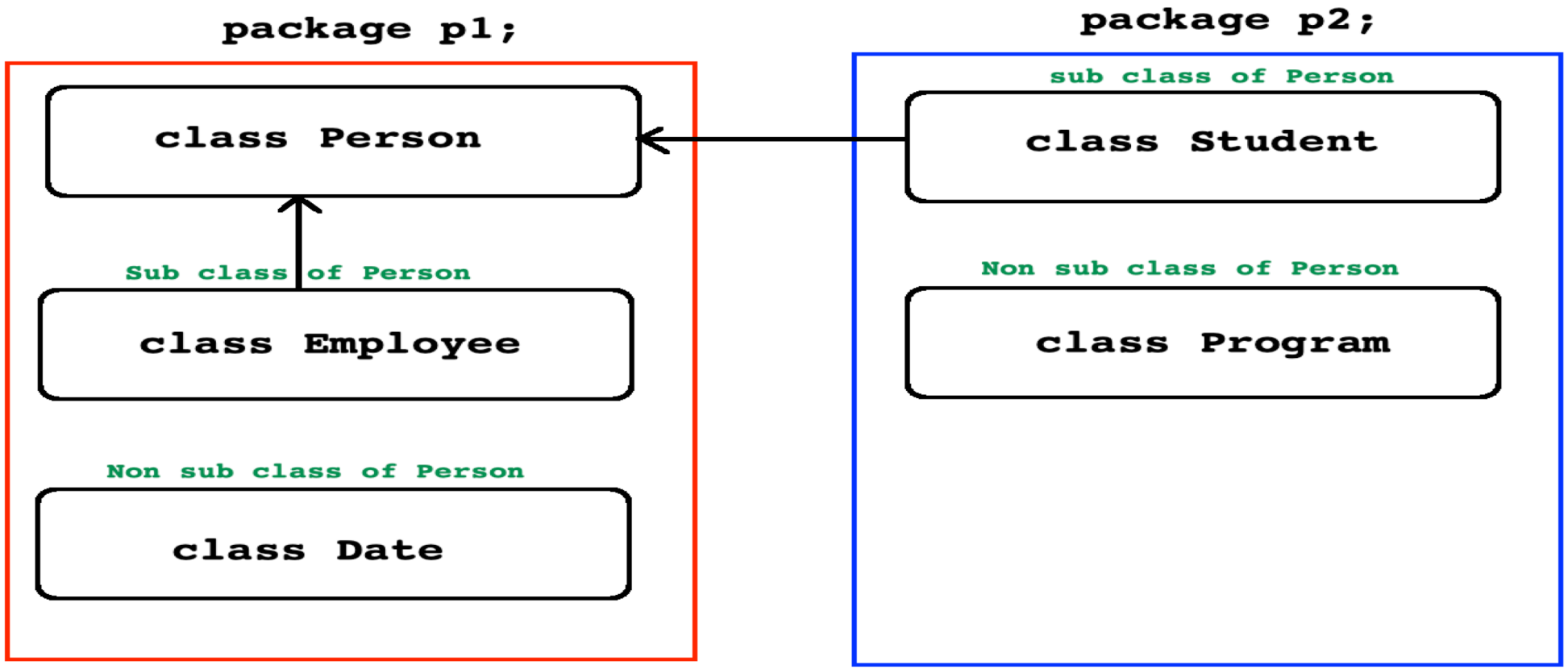
12.VOLATILE

Access Modifier

- If we want to control visibility of members of class then we should use access modifier.
- There are 4 access modifiers in Java:
    1. private
    2. package-level private / default
    3. protected
    4. public

	Same Package			Different Package	
Access Modifiers	Same class	Sub class	Non sub cass	Sub class	Non Sub class
private	A	NA	NA	NA	NA
package level private/Default	A	A	A	NA	NA
protected	A	A	A	A	NA
public	A	A	A	A	A

Access Modifier



Steps to write code in Java.

- Understand problem statement and analyze it.
- Decide class and fields for it. A variable declared inside class scope is called field.
- Create instance of a class(Instantiate Java class).
  - Process of creating instance/object from a class is called as instantiation.
- Following members of a class do not get space inside instance:
  1. Nested Type(Interface/Class/Enum)
  2. Static Field
  3. Constructor
  4. Method(static/non static)
  5. Method Parameter and Local variable.
- If we create instance of a class then only non static fields get space inside it.
- If we want to process state/value of instance/object then we should define and invoke method on instance. This process of calling method on instance is called message passing.

Class

- Consider following examples:
  1. day, month, year - related to – Date
  2. hour, minute, second - related to – Time
  3. red, green, blue - related to Color
  4. real, imag - related to – Complex
  5. xPosition, yPosition - related to Point
  6. number, type, balance - related to Account
  7. name, id, salary - related to Employee

- If we want to group related data elements together then we should use/define class in Java.

```
class Date{
    int day;        //Field
    int month;      //Field
    int year;       //Field
}
```

```
class Employee{
    String name;    //Field
    int id;         //Field
    float salary;   //Field
}
```

Class

- class is a non primitive/reference type in Java.
- If we want create object/instance of a class then it is mandatory to use new operator.
- If we create instance using new operator then it gets space on heap section.
- Only fields of the get space once per instance according to order of their declaration inside class.

Class

- **Field**
  - A variable declared inside class / class scope is called a field.
  - Field is also called as attribute or property.
- **Method**
  - A function implemented inside class/class scope is called as method.
  - Method is also called as operation, behavior or message.
- **Class**
  - Class is a collection of fields and methods.
  - Class can contain
    1. Nested Type
    2. Field
    3. Constructor
    4. Method
- **Instance**
  - In Java, Object is also called as instance.

Instantiation

- Process of creating instance/object from a class is called as instantiation.

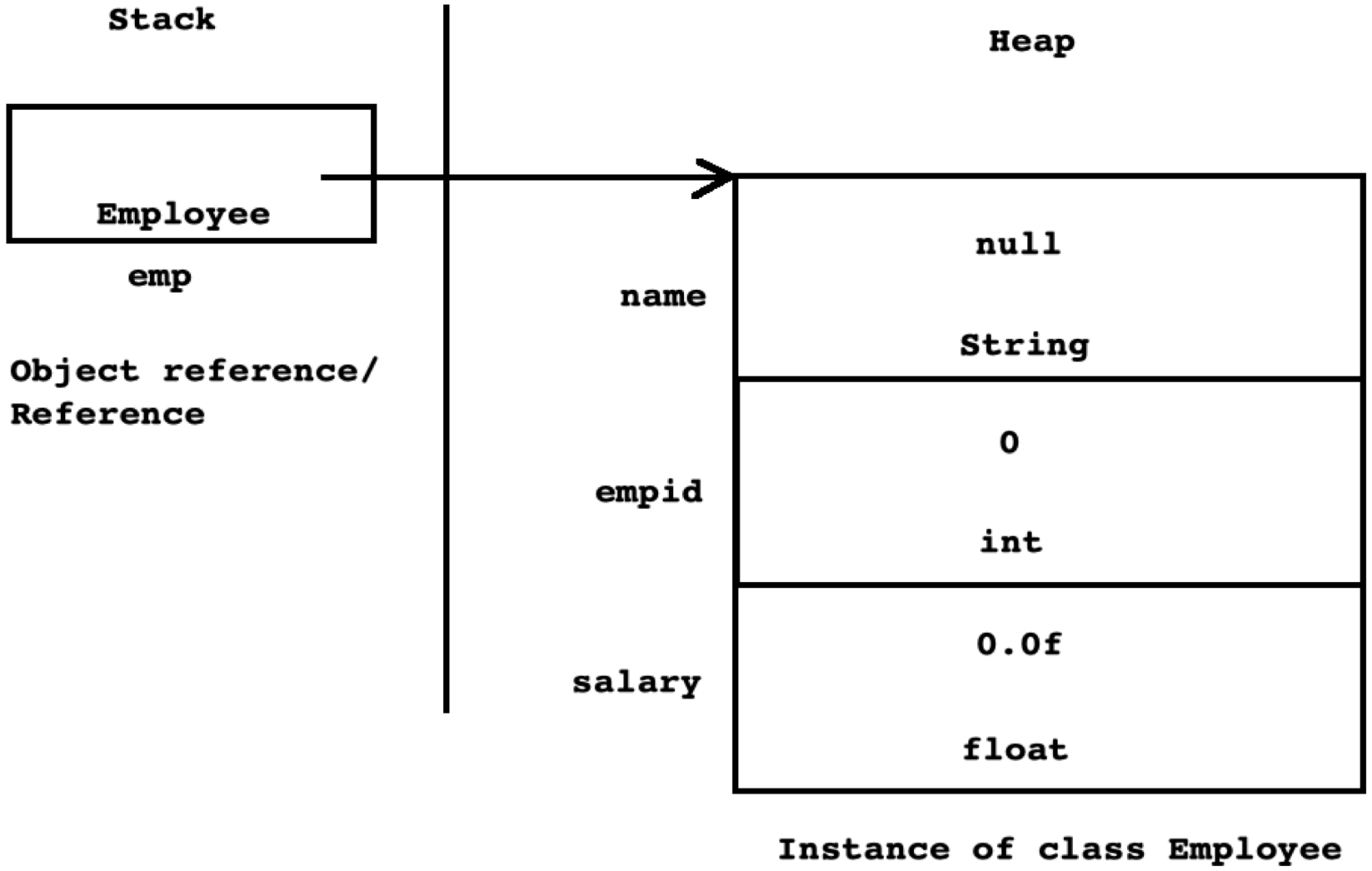
- In C programming language
  - Syntax : struct StructureName identifier_name;
  - struct Employee emp;

- In C++ programming language
  - Syntax : [class] ClassName identifier_name;
  - Employee emp;

- In Java programming language
  - Syntax : ClassName identifier_name = new ClassName( );
  - Employee emp = new Employee();

- **Every instance on heap section is anonymous.**

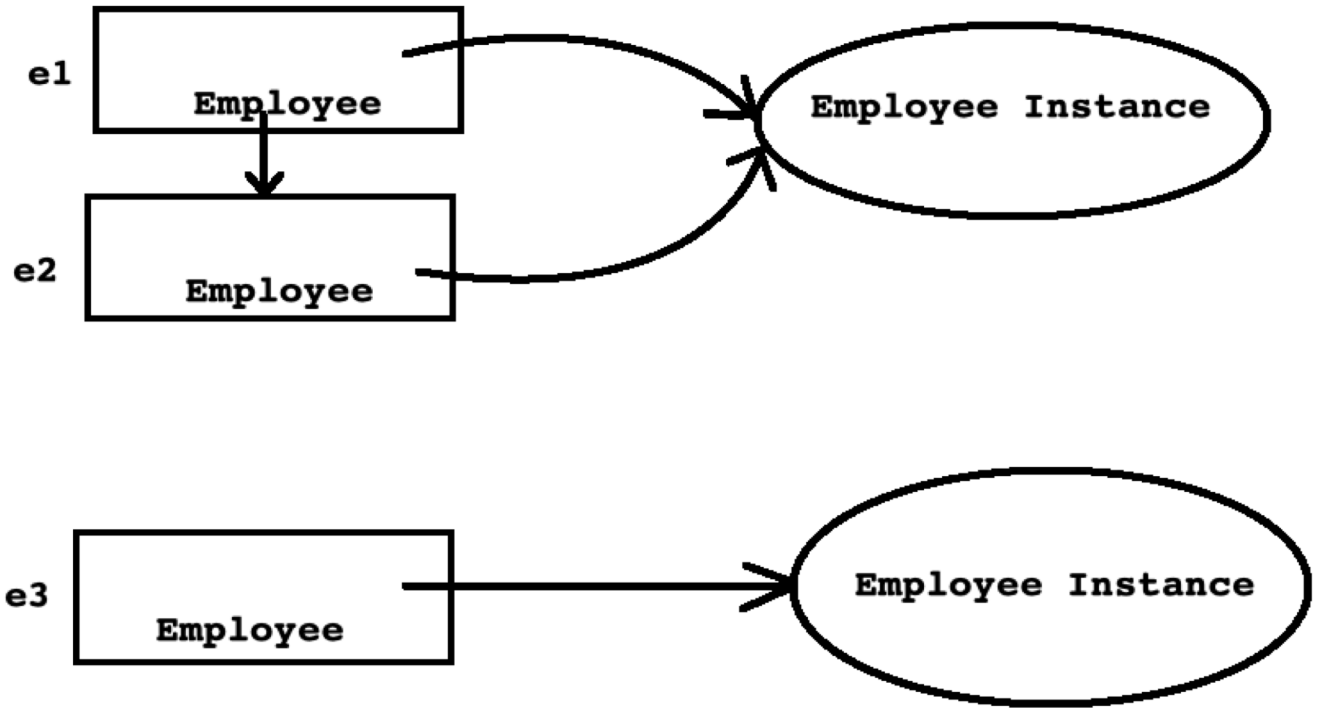
Instantiation



Instantiation

- Consider following code:

```
1. Employee e1 = new Employee();
2. Employee e2 = e1;
3. Employee e3 = new Employee();
```



this reference

- If we call non static method on instance(actually object reference) then compiler implicitly pass, reference of current/calling instance as a argument to the method implicitly. To store reference of current/calling instance, compiler implicitly declare one reference as a parameter inside method. It is called this reference.

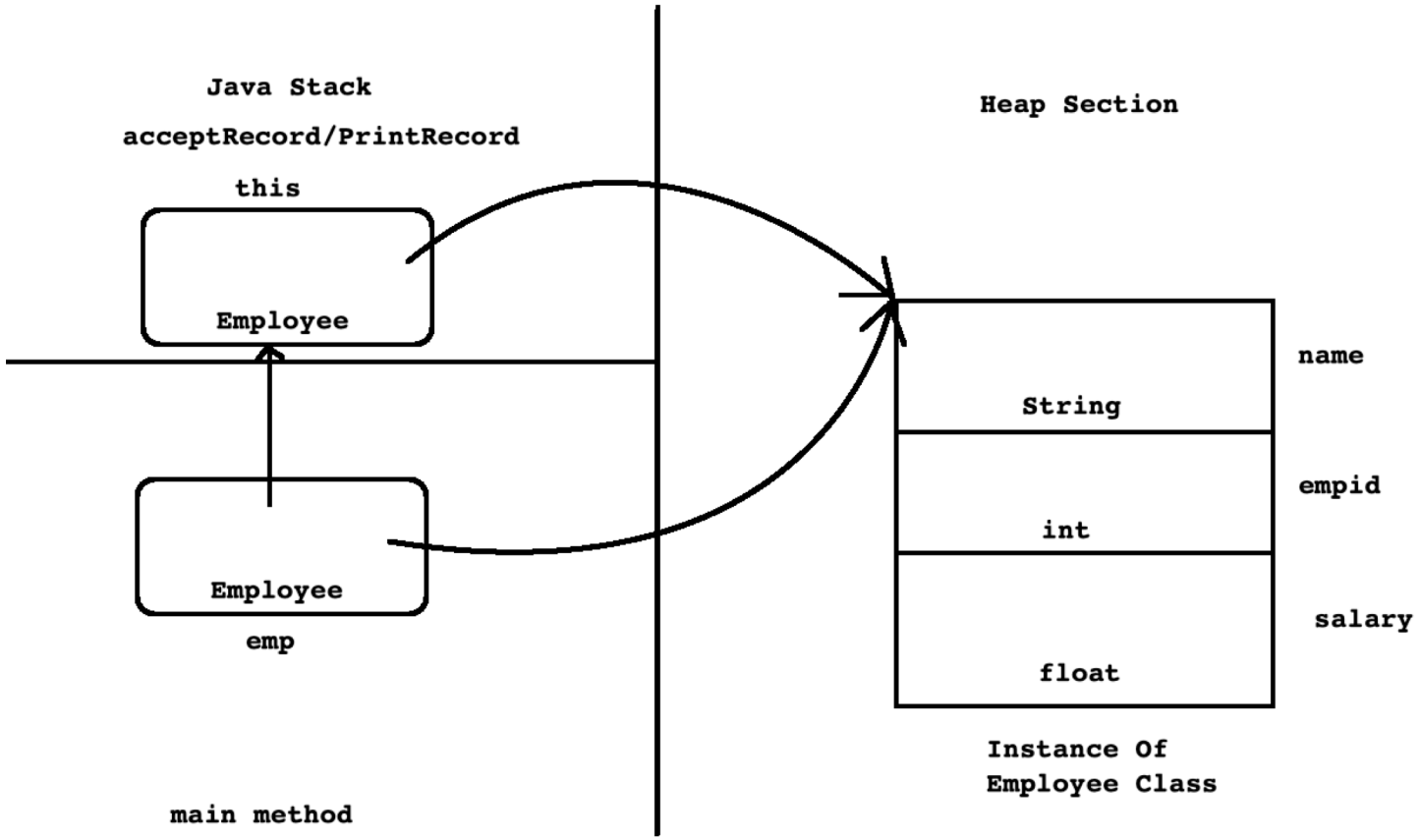
- **this is a keyword** in Java which is designed to store reference of current/calling instance.

- **Using this reference, non static fields and non static methods are communicating with each other. Hence this reference is considered as a link/connection between them.**

- Definition
  - **“this" is implicit reference variable that is available in every non static method of class which is used to store reference of current/calling instance.**

- Inside method, to access members of same class, use this keyword is optional

this reference



this reference

- If name of local variable/parameter and name of field is same then preference is always given to the local variable.

```
class Employee{
    private String name;
    private int empid;
    private float salary;
    public void initEmployee(String name, int empid, float salary ){
        this.name = name;
        this.empid = empid;
        this.salary = salary;
    }
}
```

Constructor

- If we want to initialize instance then we should define constructor inside class.
- Constructor look like method but it is not considered as method.
- It is special because:
  1. Its name is same as class name.
  2. It doesn't have any return type.
  3. It is designed to call implicitly.
  4. It gets called once per instance.
- We can not call constructor on instance explicitly

```
Employee emp = new Employee();
emp.Employee( ); //Not Ok
```

- Types of constructor:
  1. Parameterless constructor
  2. Parameterized constructor
  3. Default constructor.

Thank you