**While Executing the Constructor, If matching’s not found for Data type then it follows 3 Rules**

1. **Primitive Type Conversion(Priority 1)**
2. **Auto Boxing And up-casting (Priority 2)**
3. **Var-args (In up casted Order)(Priority 3)**

**In the Below Example Order is given as:**

**package** com.app.module;

**import** java.io.Serializable;

**public** **class** Employee {

**public** Employee(**byte** x,**byte** y) {

System.*out*.println("Order-1(never)");

}

**public** Employee(**short** x,**short** y) {

System.*out*.println("Order-2(never)");

}

**public** Employee(**int** x,**int** y) {

System.*out*.println("Order1");

}

**public** Employee(**long** x,**long** y) {

System.*out*.println("Order2");

}

**public** Employee(**float** x,**float** y) {

System.*out*.println("Order3");

}

**public** Employee(**double** x,**double** y) {

System.*out*.println("Order4");

}

**public** Employee(Integer x,Integer y) {

System.*out*.println("Order5");

}

**public** Employee(Double x,Double y) {

System.*out*.println("Order5(never)");

}

**public** Employee(Number x,Number y) {

System.*out*.println("Order6");

}

**public** Employee(Serializable x,Serializable y) {

System.*out*.println("Order7");

}

**public** Employee(Object x,Object y) {

System.*out*.println("Order8");

}

**public** Employee(Integer... integers) {

System.*out*.println("Order9");

}

**public** Employee(Number... integers ) {

System.*out*.println("Order10");

}

**public** Employee(Serializable... serializables) {

System.*out*.println("Order11");

}

**public** Employee(Object... integers) {

System.*out*.println("Order12");

}

**public** **static** **void** main(String[] args) {

//write the above constructor in any order. They are called in Specified

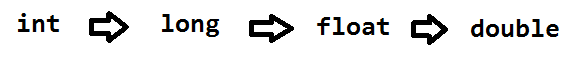
//number formats only.

Employee emp=**new** Employee(10,10);

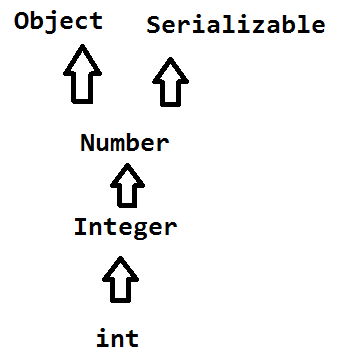
}

}

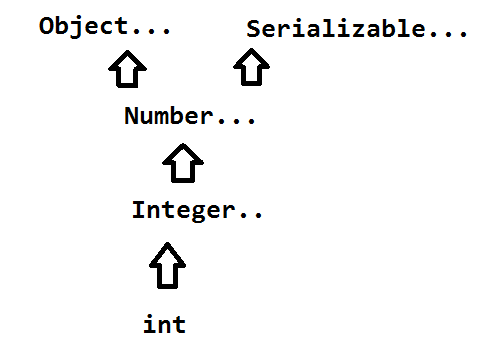
**Process 1:**



**Process 2:**

****

**Process 3:**

****

**Invalid Process:**

****

**\*\*\*\* Note: at a time type conversion and auto-boxing can never be done.**