**METHOD OVERLOADING:**

The process of developing multiple methods with same name but different arguments list is called as “METHOD OVERLOADING”.

**RULES FOR DEFINING ARGUMENT LIST:**

1. Number of arguments must be different.

**Ex:** swiggy ()

swiggy (String name)

swiggy (String name, int orderid)

2. Types(datatype) of arguments must be different.

**Ex:** add (int i, int j)

add (double d, double d1)

add (String s1, String s2)

3. Sequence or position of Arguments must be different.

**Ex:** login (String ul, int id)

login (int id, String pwd) –

* We go for method overloading **when we want to perform one task in multiple ways**
* Method Overloading is Subjected to changing of arguments.
* It is an Example of **Compile Time Polymorphism.**

**Example Program for Method Overloading:**

**public** **class** MethodOverloading

{

**public** **static** **void** Payment(String WalletType, String UID)

{

System.***out***.println("Wallet Type is:"+WalletType+" UID is:"+UID);

}

**public** **static** **void** Payment(String CardType, **long** CardNo, **int** CVVNo)

{

System.***out***.println("CardType is:"+CardType+" CardNo is:"+CardNo+" CVVNo is:"+CVVNo);

}

**public** **static** **void** Payment(String AccType, **long** AcNo, **int** UserNo, String Pwd)

{

System.***out***.println("AccType is:"+AccType+" AcNo is:"+AcNo+" UserNo is:"+UserNo+" Pwd is:"+Pwd);

}

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

{

*Payment*("Gpay","Gpay234@ybl.com");

*Payment*("Debitcard",45816432789L,420);

*Payment*("savings",35486785214L,6547523,"ABC123");

}

}

**Output:**

Wallet Type is:Gpay UID is:Gpay234@ybl.com

CardType is:Debitcard CardNo is:45816432789 CVVNo is:420

AccType is:savings AcNo is:35486785214 UserNo is:6547523 Pwd is:ABC123

--------------------------------------- \*\*\*\*\* --------------------------------------------------------------------

**Method Overriding:**

During Inheritance subclass has complete privilege to change the (method)implementation of super class, this process is known as “Method Overriding”.

(OR)

Parent Class & Child Class have same method but implementation is different this process is called as “Method Overriding”.

**Rules For Overriding:**

1. Inheritance (or) “IS A Relationship” is compulsory.
2. Method signature must be same (method name and arguments) as super class in sub class.
3. Method header must be same (access modifier, nonaccess modifier, return type)
4. Overridden (Parent Class) method should not be final.

**Example Program for Method Overriding:**

**class** Parents

{

**public** **void** car()//Overridden method

{

System.***out***.println("Blue color");

}

**public** **void** carname()

{

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

}

}

**class** Son **extends** Parents

{

**public** **void** car()//Overriding method

{

System.***out***.println("Black color");

}

**public** **void** carname()

{

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

}

}

**class** Daughter **extends** Parents

{

**public** **void** car()

{

System.***out***.println("White color");

}

**public** **void** carname()

{

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

}

}

**public** **class** MethodOverriding

{

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

{

Son S = **new** Son();

S.car();

S.carname();

Daughter D = **new** Daughter();

D.car();

D.carname();

}

}

**Output:**

Black color

BMW

White color

AUDI

--------------------------------------- \*\*\*\*\* --------------------------------------------------------------------

**Swapping Of Two Variables with Extra Variable:**

**public** **class** Swap\_with\_Variable

{

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

{

**int** a=10,b=20,c;

c=b;

b=a;

a=c;

System.***out***.println("a="+a+" b="+b);

}

}

**Output:** a=20 b=10

--------------------------------------- \*\*\*\*\* --------------------------------------------------------------------

**Swapping Of Two Variables without Extra Variable:**

**public** **class** Swap\_without\_ExtVar

{

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

{

**int** a=10,b=20;

a=a+b; //a=10+20-----a=30

b=a-b; //b=30-20-----b=10

a=a-b; //a=30-10-----a=20

System.***out***.println("a="+a+" b="+b);

}

}

**Output:** a=20 b=10