BITS Pilani
Pilani Campus

Object Oriented Programming
CS F213

J. Jennifer Ranjani
email: jennifer.ranjani@pilani.bits-pilani.ac.in
Chamber: 6121 P, NAB
Consultation: Friday 4.00 p.m. – 5.00 p.m.

Early Binding

- **Static or Early binding** links method definition and method call at compile time.
- Binding of all the static, private and final methods is done at compile-time.
- Actual object is not used for binding.
- For faster execution
- Eg. Method Overloading

BITS Pilani
Pilani Campus

‘This’ Keyword

‘this’ Keyword

- It is a reference variable that refers to the current object

- Six usage
  - ~~**this can be used to refer current class instance variable.**~~
  - ~~**this can be used to invoke current class method (implicitly)**~~
  - **this() can be used to invoke current class constructor.**
  - **this can be passed as an argument in the method call.**
  - **this can be passed as argument in the constructor call.**
  - **this can be used to return the current class instance from the method**.

this() : to invoke current class constructor

- The this() constructor call can be used to invoke the current class constructor. It is used to reuse the constructor. In other words, it is used for constructor chaining.

- Calling default constructor from parameterized constructor

- Calling parameterized constructor from default constructor

Constructor Chaining - Example

```
class Account{
int acc;
String name;
float amount;
Account(int acc, String name){
this.acc = acc;
this.name = name;}

Account(int acc, String name, float amount){
this.acc = acc;
this.name = name;
this.amount = amount;  }

void display(){
System.out.println(acc+" "+name+" "+amount);}
}
```

Constructor Chaining - Example

```
class Account{
int acc;
String name;
float amount;
Account(int acc, String name){
this.acc = acc;
this.name = name;}

Account(int acc, String name, float amount){
this(acc, name);  //reusing constructor
this.amount = amount;  }

void display(){
System.out.println(acc+" "+name+" "+amount);}
}
```

Constructor Chaining - Example

```
class TestAccount{
public static void main(String[] args){
Account a1=new Account(832345,"Ankit",5000);
a1.display();
 }}
```

```
~~Account(int acc, String name, float amount){~~
~~this.amount = amount;~~
~~this(acc, name);  //reusing constructor  }~~
```

this: to pass as an argument in the method

```
class Account{
int acc;
String name;
float amount;
Account(int acc,String name){
     this.acc = acc;
     this.name = name;
     display(this); }
void update(int act,String aname, float amt) {
     acc = act;
     name = aname;
     amount = amt;
     display(this);    }
void display(Account a){
     System.out.println(a.acc+" "+a.name+" "+a.amount);}
}
```

this: to pass as an argument in the method

```
class second{
public static void main(String[] args){
        Account a1=new Account(832345,"Ankit");
        Account a2=new Account(832345,"Shobit");
        a1.update(832346, "Aankit", 5000);  }
}
```

Output:

```
832345 Ankit 0.0
832345 Shobit 0.0
832346 Aankit 5000.0
```

this: to pass as argument in the constructor call

```
class Account{
  int acc;
  String name;

Account(int acc, String name){
                this.acc=acc;
                this.name =name;
                Branch b=new Branch(this);
                b.display();
  }
}
```

this: to pass as argument in the constructor call

```
class Branch{
  Account obj;
  int branch;

Branch(Account obj){
                 this.obj=obj;
                 this.branch = 111;
  }
  void display(){
    System.out.println(this.obj.acc+" "+this.obj.name+" "+this.branch);
  }
}
class TestAccount{
  public static void main(String args[]){
   Account a1=new Account(832345,"Ankit");     }
}
```

Output:

```
832345 Ankit 111
```

Returning Objects using this keyword

```
class Account{
int acc;
String name;
float amount;
Account(int acc,String name){
            this.acc = acc;
            this.name = name;  }
Account update(int act,String aname, float amt)   {
            acc = act;
            name = aname;
            amount = amt;
            return this;    }
void display(){
System.out.println(acc+" "+name+" "+amount);}
}
```

Returning Objects using this keyword

```
class TestAccount{
public static void main(String[] args){
Account a1=new Account(832345,"Ankit");
a1.display();
a1 = a1.update(832346, "Aankit", 5000);
a1.display();
}}
```