**Methodbinding**

* The process of connecting method signature with method implementation is called as method binding
* They are two types of method bindings present

1) Compile time binding 2) run time binding

**Example :**

{

Public static void play()

{

System.out.println(“ play cricket “);

}

}

**Compile time binding :**

The process of connecting method signature with method implementation during compile time is called as compile time binding

* Since binding happen before execution it is called as early binding
* All static and final methods will be binded during compile time

i.e during compilation only method will come to know what is its implementation

**Example :**

**class Run**

**{**

**public static void fast()**

**{**

**System.out.println(" Run fast..........");**

**}**

**}**

**class CompileTime**

**{**

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

**{**

**Run.fast();**

**}**

**}**

**Run time binding :**

The process of connecting method signature with method implementation during execution time is called as run time binding

* Since binding is happen during execution it is called as late binding
* All the non static and abstract methods will be get binded during execution time

Example :

class Play

{

public void cricket()

{

System.out.println("score 100\*..........");

}

}

class RuntimeBinding

{

public static void main(String[] args)

{

Play play = new Play();

play.cricket();

}

}