Static Binding

The execution of a program where type of object is determined/known at compile time i.e when compiler executes the code it know the type of object or class to which object belongs.

Code:

```java
public class Main {
        // This method is used to demonstrate Static Binding.
    private void print() {
        System.out.println("Hi, there ;)");
    }

    public static void main(String[] args) {
        Main mainObj = new Main();

        // Here print method is a private method of Main class
        // which is predetermined at compile time.
        // Therefore, this demostrates Static Binding.

        mainObj.print();
    }
}
```

Output:

```
PS E:\Sarath\java>  & 'C:\Program Files\Java\jdk-18\bin\java.exe' '-agentli
b:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:52482' '--e
nable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Ad
ministrator\AppData\Roaming\Code\User\workspaceStorage\2dd4318ec35c685fc913
bf090615ce48\redhat.java\jdt_ws\java_396c42a6\bin' 'Main'
Hi, there ;)
PS E:\Sarath\java>
```

Dynamic Binding

When the compiler resolves the method call binding during the execution of the program, such a process is known as Dynamic or Late Binding. In dynamic binding the compiler can't predetrmine the type of the object.

Code:

```java
public class Main {
    public static void main(String[] args) {
            // Squre class inherits from Parallelogram -> Shape class.
            // All of them have draw methods.
            // The draw method isn't predeterminded during the compile phase.
            // Because we don't know which draw mehtod are we gonna use.
            // This is determined when we create the object.
            // Therefore this shows dynamic binding.
        Shape sq = new Square();
        sq.draw();
    }
}

// Shape Class
class Shape {
    public void draw() {
        System.out.println("Drawing a shape...");
    }
}
// Parallelogram Class
class Parallelogram extends Shape {
    public void draw() {
        System.out.println("Drawing a parallelogram...");
    }
}
// Square Class
class Square extends Parallelogram {
    public void draw() {
        System.out.println("Drawing a square...");
    }
}
```

Output:

```
PS E:\Sarath\java>  e:; cd 'e:\Sarath\java'; & 'C:\Program Files\Java\jdk-1
8\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages
' '-cp' 'C:\Users\Administrator\AppData\Roaming\Code\User\workspaceStorage\
2dd4318ec35c685fc913bf090615ce48\redhat.java\jdt_ws\java_396c42a6\bin' 'Mai
n'
Drawing a square...
PS E:\Sarath\java>
```