Polymorphism

- **Method overloading by changing data type of Arguments**

```
package com.javabykiran;

class Calculate {

	void sum(int d, int e) {
		System.out.println("sum is " + (d + e));
	}

	void sum(double a, double b) {
		System.out.println("sum is " + (a + b));
	}

	public static void main(String[] args) {
		Calculate cal = new Calculate();
		cal.sum(8, 5); // sum(int a, int b) is method is called.
		cal.sum(4.6, 3.8); // sum(float a, float b) is called.
	}
}
```

- **Simple Method Overloading example**

```
package com.javabykiran;

public class Overload {

	void demo(int a) {
		System.out.println("a: " + a);
	}

	void demo(int a, int b) {
		System.out.println("a and b: " + a + "," + b);
	}

	double demo(double a) {
		System.out.println("double a: " + a);
		return a * a;
	}
```

```
        int demo(int a, int b, int c) {
                return a + b + c;
        }
}
```

```
package com.javabykiran;

public class MethodOverloading {

        public static void main(String[] args) {
                Overload Obj = new Overload();
                double result;
                int add;
                Obj.demo(10);
                Obj.demo(10, 20);
                result = Obj.demo(5.5);
                System.out.println("O/P : " + result);
                add = Obj.demo(5, 5, 5);
                System.out.println("O/P : " + add);
        }
}
```

- **Method overloading by changing no. of argument.**

```
package com.javabykiran;

class Area {

        void find(int l, int b) {
                System.out.println("Area is " + (l * b));
        }

        void find(int l, int b, int h) {
                System.out.println("Area is " + (l * b * h));
        }

        public static void main(String[] args) {
                Area ar = new Area();
                ar.find(8, 5); // find(int l, int b) is method is called.
                ar.find(4, 6, 2); // find(int l, int b,int h) is called.
        }
}
```

- **Program for method overloading**

```
package com.javabykiran;

public class Student {

	String name;
	int age; String
	email;

	public void setData(String name, int age) {
		this.name = name;
		this.age = age;
	}

	public void setData(String name, int age, String email) {
		this.name = name;
		this.age = age;
		this.email = email;
	}

	public void display() {
		System.out.println(name);
		System.out.println(age);
		System.out.println(email);
	}

	public static void main(String[] args) {
		Student s1 = new Student();
		s1.setData("Shanthi", 20);
		Student s2 = new Student();
		s2.setData("Veera", 25, "veera@candidjava.com");
	}
}
```

Examples Of Method Overriding:

```
package com.javabykiran;

public class Baseclass {

	public void method() { // Base class method
		System.out.println("I'm the method of BaseClass");
	}
}
```

```
package com.javabykiran;

public class DerivedClass extends Baseclass {
	public void method() {// Base class method
		System.out.println("I'm the method of DerivedClass");
	}
}
```

```
package com.javabykiran;

public class Override {

	public static void main(String[] args) {
		// method calling from sub class object
		DerivedClass der = new DerivedClass();
		der.method();

		// method calling from super class object
		Baseclass base = new Baseclass();
		base.method();
		Baseclass base1 = new DerivedClass();
		base1.method();
	}
}
```

- **Create a Simple Method Overriding(Dynamic Binding) in Java**

```
package com.javabykiran;

public class BindDynamic {

	protected String val;

	void display(String str) {
		val = "Base Class Fuction ".concat(str);
		System.out.println(val);
	}
}
```

```
package com.javabykiran;

class BindDynamicSubclass extends BindDynamic {

	void display(String str) {
		if (val == null) {
			str = "Derived Class Fuction ".concat(str);
			System.out.println(str);
		}
	}
}
```

```
package com.javabykiran;

class BindDynamicMainClass {

	public static void main(String args[]) {
		BindDynamicSubclass obj = new BindDynamicSubclass();
		obj.display("Called");
	}
}
```

- **Now rewrite the Code in Sub Class and check changes of output**

```
package com.javabykiran;

class BindDynSubClass extends BindDynamic {
	void display(String str) {
		super.display(str);
		if (val == null) {
			str = "Derived Class Fuction ".concat(str);
			System.out.println(str);
		}
	}
}
```

- **Complex Method Overriding (Dynamic Binding) example**

```
package com.javabykiran;

public class BindEx1 {

	String text = "BindEx1's";

	void display() {
		System.out.println(text + " function called");
	}
}
```

```
package com.javabykiran;

class BindSubclass1 extends BindEx1 {
	void display() {
		super.display();

		text  =  "SubClass1's";
		System.out.println(text + " function called");
	}
}
```

```
package com.javabykiran;

class BindSubclass2 extends BindSubclass1 {

        void display() {

                super.display();
                text  =  "SubClass2's";
                System.out.println(text + " function
                called");
        }
}
```

```
package com.javabykiran;

class BindMainclass {

        public static void main(String args[]) {
                BindSubclass2 obj = new
                BindSubclass2();obj.display();
        }
}
```

- **Method Overriding in hierarchical type**

```
package com.javabykiran;

public class Bank {

	int getRateOfInterest() {
		return 0;
	}
}
```

```
package com.javabykiran;

public class SBIBank extends Bank {
	int getRateOfInterest() {
		return 8;
	}
}
```

```
package com.javabykiran;

public class AXISBank extends Bank {
		int getRateOfInterest() {
			return 11;
		}
	}
```

```
package com.javabykiran;

public class ICICIBank extends Bank {
	int getRateOfInterest() {
		return 10;
	}
}
```

```
package com.javabykiran;

public class BankTest {
	public static void main(String[] args) {
		Bank b = new Bank();
		System.out.println("Bank Rate of Interest : " + b.getRateOfInterest() + "%");
		Bank b1 = new SBIBank();
		Bank b2 = new ICICIBank();
	Bank b3 = new AXISBank();
		System.out.println("SBI Rate of Interest : " + b1.getRateOfInterest() + "%");
		System.out.println("ICICI Rate of Interest : " + b2.getRateOfInterest() + "%");
		System.out.println("AXIS Rate of Interest : " + b3.getRateOfInterest() + "%");
	}
}
```

Homework

- Solve test on jbktest.com for polymorphism
- Read jbktutorials.com
  - https://www.jbktutorials.com/corejava/polymorphism-in-java.php#gsc.tab=0
- Read interview questions
  - https://www.jbktutorials.com/core-java-interview-questions/polymorphism-interview-questions.php#gsc.tab=0

Download

- https://drive.google.com/drive/folders/1NlzoMu_oxhmZB6JEsvMneQ75qyJ9fl3a?usp=sharing