Academic Year: 2023-24 **Semester: II** **Class: FYMCA**

Course Code: MC506 **Course Name: Java Programming**

Name: Durgesh Dilip Mandge **UCID : 2023510032**

Experiment No.3

Date: 02-04-2024

1. Define a generic class called GenAB, and a second class called GenDemo, which uses GenAB. The GenAB class should have atleast two parameters namely A and B that will be replaced by different data types whenever the object of type GenAB is created.

```
class GenAB<A, B> {
    private A a;
    private B b;

    public GenAB(A a, B b) {
        this.a = a;
        this.b = b;
    }

    public A getA() {
        return a;
    }

    public void setA(A a) {
        this.a = a;
    }

    public B getB() {
        return b;
    }

    public void setB(B b) {
        this.b = b;
    }
}
```

```java
class GenDemo {
    private GenAB<?, ?> obj;

    public GenDemo(GenAB<?, ?> obj) {
        this.obj = obj;
    }

    public void display() {
        System.out.println("A: " + obj.getA() + ", B: " + obj.getB());
    }
}

public class Main {
    public static void main(String[] args) {
        // Creating an instance of GenAB with different data types for A
and B
        GenAB<Integer, Double> genObjIntDouble = new GenAB<>(10, 3.14); //
A is Integer, B is Double
        GenAB<String, Boolean> genObjStrBool = new GenAB<>("Hello", true);
// A is String, B is Boolean

        // Using GenDemo with GenAB instances
        GenDemo demoObj1 = new GenDemo(genObjIntDouble);
        GenDemo demoObj2 = new GenDemo(genObjStrBool);

        // Displaying the values
        demoObj1.display(); // Output: A: 10, B: 3.14
        demoObj2.display(); // Output: A: Hello, B: true
    }
}
```

OUTPUT:

```
PS C:\Users\smart\Documents\SPIT-MCA\SPIT-lab\Sem 2\Java\Lab3>  & 'C:\Program Files\Java\jdk-21\bi
ges' '-cp' 'C:\Users\smart\AppData\Roaming\Code\User\workspaceStorage\712bda0fe6b9f31bac1f94902f23
'
A: 10, B: 3.14
A: Hello, B: true
PS C:\Users\smart\Documents\SPIT-MCA\SPIT-lab\Sem 2\Java\Lab3>
```

2. Write a Java program to implement simple use of Generic class and demonstrate by using various primitive data types in place of generic types.

```java
class Gen<T> {
    private T data;

    public Gen(T data) {
        this.data = data;
    }

    public T getData() {
        return data;
    }

    public void setData(T data) {
        this.data = data;
    }
}

public class Main3A {
    public static void main(String[] args) {
        // Using Gen with Integer
        Gen<Integer> genInt = new Gen<>(10);
        System.out.println("Integer data: " + genInt.getData());

        // Using Gen with Double
        Gen<Double> genDouble = new Gen<>(3.14);
        System.out.println("Double data: " + genDouble.getData());

        // Using Gen with Character
        Gen<Character> genChar = new Gen<>('A');
        System.out.println("Character data: " + genChar.getData());

        // Using Gen with Boolean
        Gen<Boolean> genBoolean = new Gen<>(true);
        System.out.println("Boolean data: " + genBoolean.getData());

        // Using Gen with Byte
        Gen<Byte> genByte = new Gen<>((byte) 8);
```

```
        System.out.println("Byte data: " + genByte.getData());

        // Using Gen with Short
        Gen<Short> genShort = new Gen<>((short) 100);
        System.out.println("Short data: " + genShort.getData());

        // Using Gen with Long
        Gen<Long> genLong = new Gen<>(1000L);
        System.out.println("Long data: " + genLong.getData());

        // Using Gen with Float
        Gen<Float> genFloat = new Gen<>(3.14159f);
        System.out.println("Float data: " + genFloat.getData());
    }
}
```

OUTPUT:

```
PS C:\Users\smart\Documents\SPIT-MCA\SPIT-lab\Sem 2\Java\Lab3>  & 'C:\Program File
ges' '-cp' 'C:\Users\smart\AppData\Roaming\Code\User\workspaceStorage\712bda0fe6b9
A'
Integer data: 10
Double data: 3.14
Character data: A
Boolean data: true
Byte data: 8
Short data: 100
Long data: 1000
Float data: 3.14159
PS C:\Users\smart\Documents\SPIT-MCA\SPIT-lab\Sem 2\Java\Lab3>
```