

LAB 3

Q1. Design a class named MyInteger. The class contains:

- An int data field named value that stores the int value represented by this object.
- A constructor that creates a MyInteger object for the specified int value.
- A getter method that returns the int value.
- The method isEven() returns true if the value in this object is even.
- The static method isEven(int) that returns true if the specified value is even.
- The static method isEven (MyInteger) that returns true if the specified value is even.
- The methods equals(int) and equals(MyInteger) that return true if the value in this object is equal to the specified value.

```
public class Myinteger {
    int data;
    public Myinteger()
    {
        data=10;
    }
    boolean isEven()
    {
        return data%2==0;
    }
    int getter()
    {
        return data;
    }
    static boolean isEven_Static(int value)
    {
        return value%2==0;
    }
    static boolean isEven(Object Myinteger)
    {
        Myinteger Myint = new Myinteger();
        return (Myint.data)%2==0;
    }
    boolean equals(int val)
    {
        return val==data;
    }
}
```

```

    }
    boolean equal(Object Myinteger,int val)
    {
        Myinteger myint= new Myinteger();
        return myint.data==val;
    }
    public static void main(String a[])
    {
        Myinteger obj1 = new Myinteger();
        System.out.println(obj1.data);
        obj1.data = 20;
        System.out.println(obj1.getter());
        System.out.println(obj1.isEven());
        System.out.println(Myinteger.isEven_Static(obj1.data));
        System.out.println(Myinteger.isEven(obj1));
        System.out.println(obj1.equals(10));
        System.out.println(obj1.equal(obj1,10));
    }
}

```

Output :

```

● (base) PS C:\Users\sansk\OneDrive\Desktop\java codes> & 'C:\Program Files\Java\jdk-11.0.2\bin\java.exe' -cp 'C:\Users\sansk\AppData\Roaming\Code\User\workspace\java codes_b6e89e30\bin' 'Myinteger'
10
20
true
true
true
false
true
○ (base) PS C:\Users\sansk\OneDrive\Desktop\java codes>

```

Q2. Create a class called Dog with an overloaded bark() method. This method should be overloaded based on various primitive data types, and print different types of barking, howling, etc., depending on which overloaded version is called. Write a main() that calls all the different versions.

Create a class without a constructor, and then create an object of that class in main() to verify that the default constructor is automatically synthesized.

Create a class with two (overloaded) constructors. Using this, call the second constructor inside the first one.

```
public class Dog2 {
    void bark()
    {
        System.out.println("The dog is barking");
    }
    void bark(int times)
    {
        System.out.println("The dog barks " + times + " times");
    }
    void bark(String type)
    {
        System.out.println("The dog is " + type);
    }
    void bark(boolean isHowling)
    {
        if(isHowling)
        {
            System.out.println("The dog is howling");
        }
        else
        {
            System.out.println("The dog is barking softly");
        }
    }
}
public class NoConstructorClass
{
}
public class OverloadedConstructorClass
{
    public OverloadedConstructorClass() {
        System.out.println("Default constructor called");
    }
}
```

```

    }

    public OverloadedConstructorClass(String message) {
        this();
        System.out.println("Overloaded constructor called with message: " +
message);
    }
}
public static void main(String args[])
{
    Dog2 myDog = new Dog2();
    myDog.bark();
    myDog.bark(3);
    myDog.bark("howling");
    myDog.bark(true);
    NoConstructorClass noConstructorObject = myDog.new NoConstructorClass();
    System.out.println("NoConstructorClass object created: " +
noConstructorObject);
    OverloadedConstructorClass overloadedObject = myDog.new
OverloadedConstructorClass("Hello, Dog!");
}
}

```

Output :

```

● (base) PS C:\Users\sansk\OneDrive\Desktop\java codes> & 'C:\Program Files\Java
ionMessages' '-cp' 'C:\Users\sansk\AppData\Roaming\Code\User\workspaceStorage\7
java codes_b6e89e30\bin' 'Dog2'
The dog is barking
The dog barks 3 times
The dog is howling
The dog is howling
NoConstructorClass object created: Dog2$NoConstructorClass@3af49f1c
Default constructor called
Overloaded constructor called with message: Hello, Dog!

```