Madan Bhandari Memorial College

Department of Computer Science and Information Technology (B.Sc.CSIT)

Ninayak Nagar, New Baneshwor, Kathmandu

Practical Sheet

Submitted By:- Utsav Acharya Sharma | Program No:- 15

Submitted To Debesh Adhikari | Lab Date:- 2080/09/22

Submission Date:- 2080/09/24 | T.U.Roll.No. :- 24109

Title: Basic Java Programs.

Class and Objects:

```
class Car{
    String make;
    String model;
    public Car(String make, String model){
            this.make = make;
            this.model = model;
    }
    public void displayDetails(){
          System.out.println("Make: "+make);
          System.out.println("Model:"+model);
    }
}
public class ClassAndObjectExample{
    public static void main(String[] args){
          Car myCar  =new car("Toyota", "Camry");
          myCar.displayDetails();
    }
}
```

Inheritance:

```
class Animal{
    void eat(){
        System.out.println("Animal is eating");
    }
}
class Dog extends Animal{
    void bark(){
        System.out.println("Dog is barking");
    }
}
public class InheritanceExample{
    public static void main (String [] args){
        Dog myDog = new Dog();
        myDog.eat();
        myDog.bark();
    }
}
```

Overloading and Overriding

```
class Calculator{
    int add(int a, int b){
        return a+b;
    }
    double add(double a, double b){
        return a+b;
    }
    void displayInfo(){
        System.out.println("Calculator Class");
    }
}
```

```
class AdvancedCalculator extends Calculator{
    @Override
     void displayInfo(){
         System.out.println("Advanced Calculator class");
     }
}
public class Overloading overriding Example{
      public static void main (String[] args){
            Calculator basicCalc = newCalculator();
            Advanced Calculator advCalc = new AdvancedCalculator();
            System.out.println("Sum (int): "+basicCalc.add(2,3));
            System.out.println ("Sum(double):"+basicCalc.add
                                                (2.5,3.5));
             basicCalc.displayInfo();
             advCalc.displayInfo();
      }
}
```

<u>Access Privileges</u>

```
class MyClass{
    private int privateVar=10;
    protected int protectedVar = 20;
    int default Var=30;
    public int publicVar=40;
    private void private Method(){
          System.out.println ("Private Method");
    }
    protected void protected Method(){
          System. out. println("Protected method");
    }
    void defaultMethod(){
          system.out.println ("Default method");
    }
    public void publicMethod(){
        system.out.println("Public method");
    }
}
public class AccessPrivleges Example{
    public static void main(String[] args){
        myClass obj = new MyClass();
        System.out.println(obj.defaultVar);
        System.out.println(obj.publicVar);
        obj.protectedMethod();
        obj.defaultMethod();
        obj.publicMethod();
    }
}
```

Interface

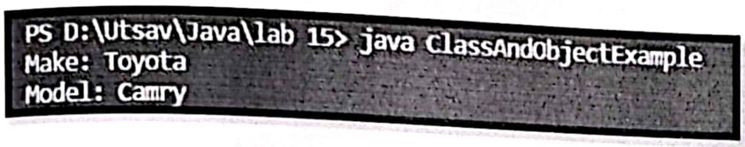
```
interface Shape{
    void draw();
}
class Circle implements Shape{
    @Override
    public void draw(){
        systen.out.println("Drawing a circle");
    }
}
public class InterfaceExample{
    public static void main(String[] args){
        Circle myCircle = new Circle();
        myCircle.draw();
    }
}
```

Inner Class

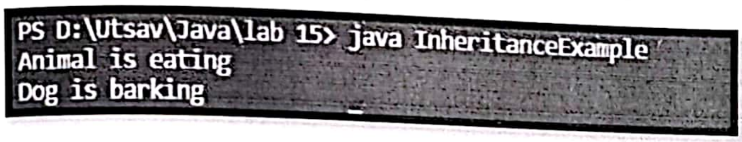
```
class Outer{
    private int outerVar = 10;
     class Inner{
        void display(){
            System.out.println("outer Variable:"+ outerVar);
        }
     }
}
public class InnerClass Example{
    public static void main (String[] args){
            Outer outerObj = new Outer();
            Outer.Inner innerObj = outerObj.new Inner();
            innerObj.display();
    }
}
```

Final Static

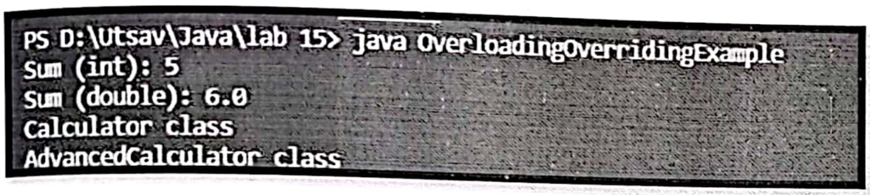
```
class Constants {
    static final double PI;
    final int SIZE =10;
    static {
        PI = 3.14159;
    }
}
public class Final Static Example{
    public static void main (String[] args){
        System.out.println("PI Value:"+constants. PI);
    }
}
```



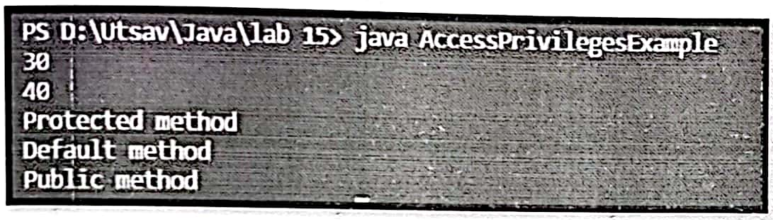
```
PS D:\Utsav\Java\lab 15> java ClassAndObjectExample
Make: Toyota
Model: Camry
```



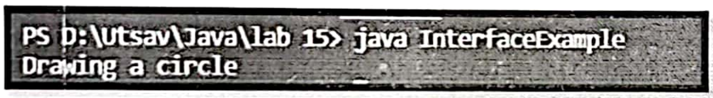
```
PS D:\Utsav\Java\lab 15> java InheritanceExample
Animal is eating
Dog is barking
```



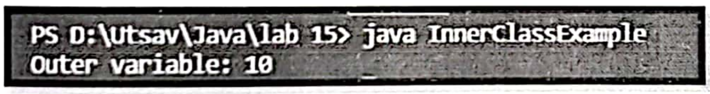
```
PS D:\Utsav\Java\lab 15> java OverloadingOverridingExample
Sum (int): 5
Sum (double): 6.0
Calculator class
AdvancedCalculator class
```



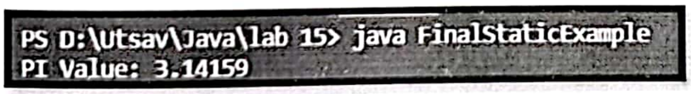
```
PS D:\Utsav\Java\lab 15> java AccessPrivilegesExample
30
40
Protected method
Default method
Public method
```



```
PS D:\Utsav\Java\lab 15> java InterfaceExample
Drawing a circle
```



```
PS D:\Utsav\Java\lab 15> java InnerClassExample
Outer variable: 10
```

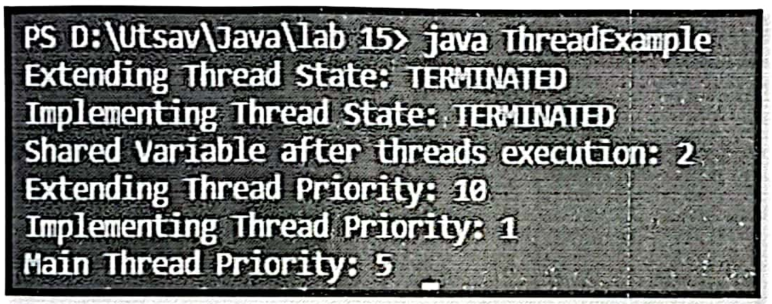


```
PS D:\Utsav\Java\lab 15> java FinalStaticExample
PI Value: 3.14159
```

✓

Thread:

```
public class ThreadExample{
    public static int sharedVariable =0;
    public static void main (String[] args){
        Thread extendingThread= new Thread(){
            public void run(){
                    sharedVariable++;
            }
        };
        extendingThread.start();
        Runnable myRunnable = () ->{
            sharedVariable++;
        };
        Thread implementingThread = new Thread(myRunnable);
        implementingThread.start();
        System.out.println("Extending Thread State: "+ extending
                                        Thread.getState());
        System.out.println("Implementing Thread State:"+ imple-
                                mentingThread.getState());
        try{
            extendingThread.join();
            implementingThread.join();
        }catch(InterruptedException e){
            e.printStackTrace();
        }
        System.out.println("Shared Variable after threads execution:
                                "+sharedVariable);
        extendingThread.setPriority(Thread.MAX_PRIORITY);
        implementingThread.setPriority(Thread.MIN_PRIORITY);
        System.out.println("Extending Thread Priority: "+ extending
                                        Thread.getPriority());
        System.out.println("Implementing Thread Priority: "+
                                implementingThread.getPriority
                                                        ());
        System.out.println("Main Thread Priority: "+ Thread.current
                                        Thread().getPriority());
        }
}
```



```
PS D:\Utsav\Java\lab 15> java ThreadExample
Extending Thread State: TERMINATED
Implementing Thread State: TERMINATED
Shared Variable after threads execution: 2
Extending Thread Priority: 10
Implementing Thread Priority: 1
Main Thread Priority: 5
```

✓