

**KHAN MOHD. OWAIS RAZA****20BCD7138**

**Q.1] a)** Create an interface named Turner, with a single method named turn(). Create a class named Leaf that implements turn() to display Changing colors. Create a class named Page that implements turn() to display Going to the next page. Create a class named Pancake that implements turn() to display Flipping. Write an application named DemoTurners that creates one object of each of these class types and demonstrates the turn() method for each class. Save the files as Turner.java, Leaf.java, Page.java, Pancake.java, and DemoTurners.java.

Turner.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public interface Turner {
    public abstract void turn();
}
```

Leaf.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class Leaf implements Turner {
    public void turn() {
        System.out.println("CHANGING COLOURS");
    }
}
```

Page.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSED2005 LAB-5 */
public class Page implements Turner{
    public void turn(){
        System.out.println("GOING TO NEXT PAGE");
    }
}
```

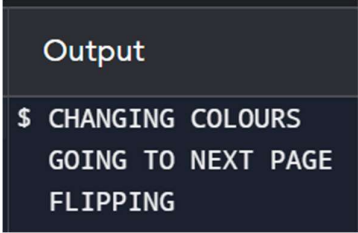
Pancake.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class Pancake implements Turner {
    public void turn(){
        System.out.println("FLIPPING");
    }
}
```

DemoTurners.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class DemoTurners
{
    public static void main(String[] args)
    {
        Leaf aLeaf = new Leaf();
        Page aPage = new Page();
        Pancake aCake = new Pancake();
        aLeaf.turn();
        aPage.turn();
        aCake.turn();
    }
}
```

Output :-



```
Output
$ CHANGING COLOURS
GOING TO NEXT PAGE
FLIPPING
```

**Q.1] b)** Think of two more objects that use turn(), create classes for them, and then add objects to the DemoTurners application, renaming it DemoTurners2. java. Save the files, using the names of new objects that use turn().

DemoTurners2.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class DemoTurners2
{
    public static void main(String[] args)
    {
        Leaf aLeaf = new Leaf();
        Page aPage = new Page();
        Pancake aCake = new Pancake();
        Biscuit aBiscuit = new Biscuit();
        Cookie aCookie = new Cookie();
        aLeaf.turn();
        aPage.turn();
        aCake.turn();
        aBiscuit.turn();
        aCookie.turn();
    }
}
```

Output :-

```
Output
$ CHANGING COLOURS
GOING TO NEXT PAGE
FLIPPING
MAKING A TURN
TURN RIGHT
```

**Q.1] c)** Apply Dynamic method dispatch to show the power of it and name the class as DemoTurners3.

DemoTurners3.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class DemoTurners3
{
    public static void main(String[] args)
    {
        Turner Obj1 = new Leaf();
        Turner Obj2 = new Page();
        Turner Obj3 = new Pancake();
        Turner Obj4 = new Biscuit();
        Turner Obj5 = new Cookie();
        Obj1.turn();
        Obj2.turn();
        Obj3.turn();
        Obj4.turn();
        Obj5.turn();
    }
}
```

Output :-

```
Output
$ CHANGING COLOURS
GOING TO NEXT PAGE
FLIPPING
MAKING A TURN
TURN RIGHT
```

**Q.2]** Modify question (3) of Lab exercise 4, adding an interface called SidedObject that contains a method called displaySides(); this method displays the number of sides the object possesses. Modify the GeometricFigure subclasses to include the use of the interface to display the number of sides of the figure. Create an application that demonstrates the use of both subclasses. Save the files as GeometricFigure2.java, Square2.java, Triangle2.java, SidedObject.java, and UseGeometric2.java.

GeometricFigure2.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public abstract class GeometricFigure2 implements SidedObject
{
    protected int height;
    protected int width;
    protected String figure;
    public GeometricFigure2(int h, int w, String f)
    {
        height = h;
        width = w;
        figure = f;
    }
    public int getHeight()
    {
        return height;
    }
    public int getWidth()
    {
        return width;
    }
    public String getFigure()
    {
        return figure;
    }
    public abstract double figureArea(int h, int w);
}
```

Square2.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class Square2 extends GeometricFigure2
{
    private double area;
    public Square2(int w, int h, String f)
    {
        super(w, h, f);
    }
    public double figureArea(int w, int h)
    {
        area = (double)(w * h);
        return area;
    }
    public void displaySides()
    {
        System.out.println("This figure has four sides");
    }
}
```

Triangle2.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class Triangle2 extends GeometricFigure2
{
    private double area;
    public Triangle2(int w, int h, String f)
    {
        super(w, h, f);
    }
    public double figureArea(int w, int h)
    {
        area = (double)(w * h * 0.5);
        return area;
    }
    public void displaySides()
    {
        System.out.println("This figure has three sides");
    }
}
```

SidedObject.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public interface SidedObject
{
    public void displaySides();
}
```

UseGeometric2.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class UseGeometric2
{
    public static void main(String[] args)
    {
        double area;
        int height;
        int width;
        String figureType;
        GeometricFigure2[] figure = new GeometricFigure2[4];
        Square2 f0 = new Square2(5, 5, "Square");
        Triangle2 f1 = new Triangle2(5, 5, "Triangle");
        Square2 f2 = new Square2(10, 10, "Square");
        Triangle2 f3 = new Triangle2(10, 10, "Triangle");
        figure[0] = f0;
        figure[1] = f1;
        figure[2] = f2;
        figure[3] = f3;
        for(int i = 0; i < figure.length; i++)
        {
            height=figure[i].getHeight();
            width=figure[i].getWidth();
            figureType=figure[i].getFigure();
            area=figure[i].figureArea(height, width);
            System.out.println("The " + figureType + " with height = "
                               + height + " and with width = " + width +
                               " has an area of " + area);
            figure[i].displaySides();
        }
    }
}
```

Output :-

Output	Clear
\$ The Square with height = 5 and with width = 5 has area of 25.0 The Triangle with height = 5 and with width = 5 has area of 12.5 The Square with height = 10 and with width = 10 has area of 100.0 The Triangle with height = 10 and with width = 10 has area of 50.0	



**Q.3]** Sanchez Construction Loan Co. makes loans of up to \$100,000 for construction projects. There are two categories of Loans—those to businesses and those to individual applicants.

Write an application that tracks all new construction loans. The application also must calculate the total amount owed at the due date (original loan amount + loan fee).

The application should include the following classes:

- **Loan** – A public abstract class that implements the LoanConstants interface. A Loan includes a loan number, customer last name, amount of loan, interest rate, and term. The constructor requires data for each of the fields except interest rate. Do not allow loan amounts greater than \$100,000. Force any loan term that is not one of the three defined in the LoanConstants class to a short-term, 1-year loan. Create a toString() method that displays all the loan data.
- **LoanConstants** – A public interface class. LoanConstants includes constant values for short term (1 year), medium-term (3 years), and long-term (5 years) loans. It also contains constants for the company name and the maximum loan amount.
- **BusinessLoan** – A public class that extends Loan. The BusinessLoan constructor sets the interest rate to 1% more than the current prime interest rate.
- **PersonalLoan** – A public class that extends Loan. The PersonalLoan constructor sets the interest rate to 2% more than the current prime interest rate.
- **CreateLoans** – An application that creates an array of five Loans. Prompt the user for the current prime interest rate. Then, in a loop, prompt the user for a loan type and all relevant information for that loan. Store the created Loan objects in the array. When data entry is complete, display all the loans.

Save the files as Loan.java, LoanConstants.java, BusinessLoan.java, PersonalLoan.java, and CreateLoans.java

LoanConstants.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSED2005 LAB-5 */
public interface LoanConstants
{
    public static final int MAXLOAN = 100000;
    public static final int SHORT_TERM = 1;
    public static final int MEDIUM_TERM = 3;
    public static final int LONG_TERM = 5;
    public static final String COMPANY = "Sanchez Construction Loan Ltd.";
}
```

BusinessLoan.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class BusinessLoan extends Loan
{
    public BusinessLoan(int num, String name, double amt, int yrs, double prime)
    {
        super(num, name, amt, yrs);
        rate = prime + 0.01;
    }
}
```

Loan.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSED2005 LAB-5 */
public abstract class Loan implements LoanConstants
{
    protected int loanNum;
    protected String lastName;
    protected double amount;
    protected double rate;
    protected int term;
    public Loan(int num, String name, double amt, int yrs)
    {
        loanNum = num;
        lastName = name;
        if(amt > MAXLOAN)
            amount = MAXLOAN;
        else
            amount = amt;
        if(yrs == SHORT_TERM || yrs == MEDIUM_TERM || yrs == LONG_TERM)
            term = yrs;
        else
            term = SHORT_TERM;
    }
    public String toString()
    {
        int r = (int)(rate * 100);
        return "Loan #" + loanNum + "    Name: " +
            lastName + " $" + amount + "\n for " +
            term + " year(s) at " + r + "% interest";
    }
    public String toString()
    {
        int r = (int)(rate * 100);
        return "Loan #" + loanNum + "    Name: " +
            lastName + " $" + amount + "\n for " +
            term + " year(s) at " + r + "% interest";
    }
    public boolean equals(Loan loan) {
        boolean returnVal = false;
        if(loanNum == loan.loanNum)
            returnVal = true;
        return returnVal;
    }
}
```

PersonalLoan.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
public class PersonalLoan extends Loan
{
    public PersonalLoan(int num, String name, double amt, int yrs, double prime)
    {
        super(num, name, amt, yrs);
        rate = prime + 0.02;
    }
}
```



CreateLoans.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-5 */
import java.util.*;
public class CreateLoans implements LoanConstants
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        final int MAX = 5;
        Loan[] loan = new Loan[MAX];
        int x;
        int num;
        String name;
        double amt, rate;
        int loanType;
        int term;
        String inStr, outString = COMPANY;
        System.out.println("Welcome to "+COMPANY +"\nEnter the current prime
interest rate : ");
        inStr = sc.nextLine();
        rate = Double.parseDouble(inStr);
        for(x = 0; x < MAX; ++x)
        {
            System.out.println("Enter the type of loan :");
            inStr = sc.nextLine();
            loanType = Integer.parseInt(inStr);
            System.out.println( "Enter account number : ");
            inStr = sc.nextLine();
            num = Integer.parseInt(inStr);
            System.out.println("Enter the name : ");
            name = sc.nextLine();
            System.out.println("Enter loan amount : ");
            inStr = sc.nextLine();
        }
    }
}
```

```

        amt = Double.parseDouble(inStr);
        System.out.println("Enter the term: ");
        inStr = sc.nextLine();
        term = Integer.parseInt(inStr);
        if(loanType == 1)
        {
            BusinessLoan temp = new
                BusinessLoan(num, name, amt, term, rate);
            loan[x] = temp;
        }
        else
        {
            PersonalLoan temp = new PersonalLoan(num, name, amt, term, rate);
            loan[x] = temp;
        }
    }
    for(x = 0; x < MAX; ++x)
        outString += loan[x].toString() + "\n";
    System.out.println(outString);
}
}

```

Output :-

```

Output
$ Welcome to Sanchez Construction Loan Ltd.
Enter the current prime interest rate :
5
Enter the loan number :
123
Enter the type of loan :
Business Loan
Enter account number :
1234567890
Enter the name :
Sebastian Schultz
Enter the loan amount :
7500
Enter the term :
3
Loan #123 Name : Sebastian Schultz $7500
for 3 year(s) at 5% interest

```

## Output

```
$ Welcome to Sanchez Construction Loan Ltd.  
Enter the current prime interest rate :  
2  
Enter the loan number :  
101  
Enter the type of loan :  
Personal Loan  
Enter account number :  
11233455677  
Enter the name :  
Charles Alexander  
Enter loan amount :  
4000  
Enter the term :  
2  
Loan #101 Name: Charles Alexander $4000  
for 2 year(s) at 2% interest
```

-----X-----X-----

