Book database

Date / /
Page

```
1) import java.util.Scanner;

class Books
{
  String name;
  String author;
  int price;
  int numpages;

Books(){
}
Books(String name, String author, int
price, int numpages)
{
  this.name = name;
  this.author = author;
  this.price = price;
  this.numpages = numpages;
}

public string toString()
{
  String name, author, price, numpages;

name = "Book name:" + this.name + "\n";
author = "Author name:" + this.author
                        + "\n";
price = "price:" + this.price + "\n";
numpages = "number of pages:" + this.numpages+
                        "\n";
return name + autha + price + numPages;
}
}
```

```
class Main
{
  public static void main(String args[])
  {
    Scanner sc = new Scanner(System.in);
    int n;

    String name;
    String author;
    int price,
    int numpages;
    System.out.Print("Enter the number of
                                books:");
    n = sc.next Int();

    Books b[];
    b = new Books[n];

    for (int i=0; i<n; i++)
    {
      System.out.println("Book"+(i+1)+":");
      System.out.print("Enter name of book
      name = s.next();
      System.out.print("Enter author:");
      author = sc.next();
    }

    for (int i=0; i<n; i++)
      System.out.println("Book "+(i+1)+":"
                                  +b[i]);
    }
}
```

Output

Enter the number of book : 2

Book 1:
Enter the name of book : Jungle book
Enter the author of book : Rudyard
Enter the price of book : 1000
Enter the number of pages : 500

Book 2:
Enter the name of book : Tale of akbar & birbal
Enter author : birbal
Enter price : 500
Enter no of pages : 500

2) import java.util.*;
class student {
String USN;
String name;
double[] marks = new double[6];

void inputDetails() {
Scanner sc = new Scanner(System.i

System.out.print("Enter USN");
USN = sc.nextLine();

System.out.println("Enter marks fo
subjects");

```
        System.out.print("Subject" + (i+1) + ":");
        marks[i] = sc.nextDouble();
    }
}

double calculatePercentage(){
    double totalMarks = 0;
    for(double mark : marks){
        totalMarks += mark;
    }
    return (totalMarks/6);
}

void displayDetails(){
    System.out.println("Student details:");
    System.out.println("USN: " + usn);
    System.out.println("percentage: " +
        calculatePercentage() + "%");
}
}

Public class Main{
    Public static void main(String[]
        args){

    Scanner sc = new Scanner(System.in);
    System.out.print("Enter no of students:");
    int numberOfStudents = sc.nextInt();

Student[] students = new Student
    [numberOfStudents];
```

```
for(int i=0; i<number of students; i++)
{
System.out.println("\n Enter details,
for student "+(i+1)+":");
students[i] = new student();
Students[i] = input details();
}

for( student student : students){
    student.displayDetails();
}
}
```

Output

Enter the number of students: 2
Enter details for student 1:
Enter USN: 295
Enter name: sulaiman
Enter marks for 6 subjects:
subject 1: 90
subject 2: 94
subject 3: 98
Subject 4: 97
subject 5: 96
Subject 6: 99
Enter details for Student 2: Enter USN
: 292
Enter name sufian
Enter marks for 6 subjects
subject 1: 65
subject 2: 6
subject: 43

Subject 4: 34
Subject 5: 55
Subject 6: 67
Student detail,
USN: 295
name: sudaiman
percentage: 96.5%.

Student detail,
USN: 292
Name: sufian
Percentag: 45.0%.

Lab-1

1) Print 'Hello world'

→ class example
{
public static void main (String a[])
{
System.out.println("Hello world");
}
}

Output:-
Hello world

2) Addition

→ class Example
{
public static void main (String a[])
int x = 10;
int y = 20

```
        int sum = x+y;
        System.out.println(sum);
    }
}
```

Output :- 30

3) Subtraction

→
```
class example
{
    public static void main(String a[])
    {
        int x = 10, y = 5, diff;
        diff = x - y
        System.out.println(diff);
    }
}
```

Output :- 5

4) Multiplication

→
```
class Example
{
    public static void main(String a[])
    {
        int x = 5, y = 6 prod;
        prod = x * y;
        System.out.println(prod);
    }
}
```

Output :- 30

5) division

→
```
class Example
{
    public static void main(String a[])
    int x = 10, y = 5, Q;
    Q = x / y;
    System.out.println(Q);
    }
}
```

6] check if no is prime
→ class Example
{
public static void main (String a[])
{
int a = 21, flag = 0
for (int i = 2; i <= a/2; i++)
if (a%i == 0)
{
flag = 1;
break;
}
if (flag == 0) {
System.out.println ("prime no");
}
}

Output
29 prime no

7) fibonacci series
→ class Example {
{
public static void main (String a[])
{
int a = 10, Term1 = 0 Term2 = 1
.Sout ("fib " + a + " term");
for (int i = 1; i < a; i++)
{
Sout (Term1 + " ");
int sum3 = sum1 + sum2;
sum1 = sum2;
sum2 = sum3;
}
}
}
Output :- 0 1, 1, 2, 3 5 . -

8) Root of quadratic equation

→ class Example

```
{
  public static void main(String a[])
  {
    double a=1, b=7, c=10;
    double root1, root2;
    double determinant = b*b-4*a*c;
    if (determinant >0)
    {
      root1 = (-b+Math.sqrt(determinant))/(2*a);
      root2 = (-b-Math.sqrt(determinant))/(2*a);
      Sout("root1 = "+root1+" and root2 = "+root2);
    }
    else if (determinant ==0)
      root1 = root2 = -b/(2*a);
      Sout("root1 = root2 = "+root1);
    }
    else
    {
      double real = -b/(2*a)
      double imaginary = Math.sqrt(-determinant)/(2*a);
      Sout("root1 = "+real+"+"+imaginary+"i");
      Sout("root2 = "+real+"-"+imaginary+"i");
    }
  }
}
```

Output:

root1 = 2 & root2 = 5

Lab-2

1) write a program to overload the method print that prints sum of a natural no when a var is passed & print the prime in a given range when 2 point a passed

```
class overload {
    void print(int n) {
        int sum = 0;
        for(int i=0; i<=n; i++){
            sum = sum+i;
        }
        sout("sum of "+n+"natural numbers" is + sum);
    }
    void print (int n, int n) {
        sout("Prime no int range a")
        for(int i=n; i<=n i++){
            int flag=0;
            for(int j=2; j<=i/2; j++){
                if(i%j==0){
                    flag=1;
                    break;
                }
            }
            if(flag==0)
                sout(i);
        }
    }
}
class overload {
    psvm {
        overload o=new overload();
        o.print(5);
    }
}
```

output
sum of 5 natural no is 15
prime numbers in the range are
7, 11, 13

2) write a java program to create a class Grocery that has the var c_name & c_phno. create a method to accept 3 prices to specify quantity of dal quantity of pulses & quantity of sugar.

→ class Grocery {
    String c_name;
    String c_pp;
    double total;
    Grocery (String c_name, String c_ph) {
        this.c_name = c_name;
        this.c_ph = c_ph;
    }
void calc (double q_dal, double q_pulses,
            double q_sugar) {
total = q_dal * 100 + q_pulses * 80 + q_sug
}
void display ()
{
    sout ("Name" + " " + "Phno" + " ");
    sout (c_name " " + total);
    sout ();
}
}

```
class GrDemo {
  psvm {
    Grocery g1 = new Grocery("ram", 2, 1);
    Grocery g2 = new Grocery("shae", 1, 1);
    g1.calc(2, 2, 1);
    g1.display();
    g2.calc(3, 5, 2);
    g2.display();
  }
}
```

Output

Name	phone	total
Rane	806	410
shae	721	800

3) Write a java program to calc roots of quadratic equation. Use app. meth to take input & calc roots.

→
```
import java.util.Scanner;
class Quad {
  int a, b, c;
  double root1, root2, d;
  Scanner s = new Scanner(System.in);
  void input()
  {
    Sout("Quadratic eqn is ax^2+bx+c");
    sout("enter a");
    a = s.nextInt();
    Sout("enter b");
    b = s.nextInt();
    Sout("enter c");
    c = s.nextInt();
  }
```

```
void discriminant(){
    d = (b*b) - (4*a*c);
}
void calculateRoots(){
    if (d>0)
    {
    sout("Roots are real & equal");
    root1 = (-b + Math.sqrt(d))/2*a;
    root2 = (-b - Math.sqrt(d))/2*a;
    sout("First root is:" + root1);
    Sout("Second root is:" + root2);
    }
    else if (d==0)
    {
    sout("Roots are real and equal");
    root1 = (-b + Math.sqrt(d))/(2*a);
    System.out.println("Root:" + root1);
    }
    else
    {
    System.out.println("No real solution
roots are imaginary");
    double real = -b/(2*a);
    double imaginary = Math.sqrt(-d)/(2*a)
    Sout("The equation has two");
    Complex roots:" + real + "+" imaginar+"
    and + real + "-" + imaginary + "i");
    }
}
class main{
    psvm {
    Quad q = new Quad();
    q.input();
    q.discriminant();
```

}
    scale roots();
}

output
Enter a 1
-4
4

roots are equal x=2

Lab-4

1. Develop a java program to create an abstract named Shape that contains two integers & an empty method named printArea(). Provide 3 classes named rectangle, triangle & circle such that each one of the classes contain the method printArea() that print the area of the given shape.

→
```
abstract class shape {
    int length;
    int width;
    public shape(int length, int width){
        this.length = length;
        this.width = width;
}
public abstract void printArea();
}
class Rectangle extends shape {
    public Rectangle(int length, int width){
        super(length, width);
    }
```

```
public void printArea() {
    int area = length * width;
    System.out.println("Rectangle Area: " + area);
}
}

class Triangle extends Shape {
    public Triangle(int length, int width) {
        super(length, width);
    }
    public void printArea() {
        double area = 0.5 * length * width;
        System.out.println("Triangle area: " + area);
    }
}

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }
    public void printArea() {
        double area = Math.PI * length * length;
        System.out.println("Circle Area: " + area);
    }
}

public class Main {
    psvm {
        Rectangle r = new Rectangle(5, 10);
        r.printArea();

        Triangle t = new Triangle(4, 6);
        t.printArea();

        Circle c = new Circle(3);
        c.printArea();
    }
}
```

Output

Rectangle Area : 50.00

Triangle Area : 12.00

circle area : 28.26

2. Develop a Java program to create a class bank that maintains two kinds of account for its customers, one called saving & the other current account. Saving account provides CI & withdrawal facilities but no cheque book facility. Current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance & if balance falls below this is imposed. Create classes Account store cust name, acc no. Include necc methods

(a) Accept deposit from custom & update balance

(b) display balance

(c) compute & deposit interest

(d) permit withdrawal & update balance. Check for min balance, impose penalty if necessary & update the balance.

```
import java.util.*;

class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    public Account(String customerName,
    int accountNumber, String accountType,
    double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance;
    }
    public void deposit(double amount){
        balance += amount;
        System.out.println("Deposit successful.
    Updated balance: " + balance);
    }
    public void displayBalance(){
        System.out.println("Account no: " + accountNum);
        System.out.println("Balance: " + balance);
    }
}
class SavAcct extends Account{
    double interestRate;
    public SavAcct(String custName,
    accNum, double balance){
        super(customerName, accNumber, "savi
        balance);
    this.interestRate = 0.05;
    }
```

```
public void deposite(int amt){
    double int amt = balance + int amt;
    balance = int;
    sout(balance);
}
public void withdraw(double amt){
    if(balance >= amt){
        balance -= amt;
        sout("withdrawal successfull" + balance);
    }
    else{
        sout("Insufficient funds");
    }
}

class CurAcct extends Account{
    double min balance;
    double service charge;

public CurAcct(String cutomerName,
int accountNumber, double balance,
double minbalance, double servicecharge){
    super(cutname, balance);
}
public void withdraw(double amount){
    if(balance - amount >= minbalance){
        balance -= amount;
        sout("withsucc" + balance);
    } else {
        sout("Insufficient" + servicecharge);
        balance -= srvicchag;
        sout("updated balance" + balance);
    }
}
```

```
public class bank {
    psvm {
        Scanner sc = new Scanner(System.in);
int choice, name, accn, bal, minb,
while(1) {                          sc
    sout("menu");
    sout("press 1 for savings");
    sout("press 2 for curr acc");
    switch(choice) {
    case 1: sout("enter name");
    name = sc.nextLine();
    sout("enter accn");
    accn = sc.nextInt();
    sout("enter balance");
    bal = sc.nextInt();
    SavAcct save = new SavAcct("name,
                    accn, bal);
    save.deposit(~~bal~~ 1000);
    save.displaybalance();
    save.depositIntrst();
    save.Account withdraw(200);
    break;

case 2: sout("enter name");
name = sc.nextLine();
sout("enter Accno");
Accn = sc.nextInt();
sout("enter balance");
bal = sc.nextInt();
sout("enter minbal")
minb = sc.nextInt();
sout("enter sc")
SC = sc.nextInt();
```

```
        }
    }
```

Lab-5

② class WrongAge Exception extends Exception {
    public WrongAge Exception (String message) {
        super (message);
    }
}

class father {
    int age;
    Public father (int age) throws
WrongAge Exception {
        if (age < 0) {
            throw new WrongAge Exception
                ("Age cannot be negative");
        }
        this.age = age;
    }
}

class son extends father {
    int sonAge;
    public Son (int fatherage, int
sonAge) throws WrongAge Exception
    {
    super (fatherage);

```
if (sonage >= fatherage) {
    throw new WrongAge Exception("Son's
age can't be greater or equal to the
father's age");
this.sonage = sonAge;
}

public class InheritanceExceptionHandling
{ public static void main(String[]
    args) {
try { int fatherAge = Integer.parseInt
                    (args[0]);
int sonAge = Integer.parseInt(args[1]);
Son son = new Son(fatherage, sonAge);
Sout (" Father's age: " + sonAge);
} catch (NumberFormatException e) {
    sout (" plese enter valid ages as
        integers."); }
catch (WrongAgeException e) {
    sout (" Error: " + e.getMessage());
}
}
```

(3) class DisplayThread extends Thread {
private final String message;
private final int interval;

public displayThread (String message,
int interval) {
this.message = message;
this.interval = interval;
}

```
public void run() {
try {
  while (true) {
    sout(message);
    Thread.sleep(interval * 1000);
  }
} catch (InterruptedException e) {
    e.printStackTrace();
}
}

public class TwoThreadsDemo {
  psvm {
  Display thread thread1 = new Display
  Thread("BMS College of Eng", 10);
  Display thread thread2 = new Display
   Thread("CSE", 2);

  thread1.start();
  thread2.start();
  }
}
```

Output – (3)

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
:
:
: