**Activity – 1**

# **Source Code:**

public class Account {

    int balance;

    String cnic;

    int yearofOpening ;

    public void Account(){

        // default Constructor

    }

    public void Account(int bal){

        //One-Argument Constructor

        if(bal>0){

            balance = bal;

        }

    }

    public void Account(int bal, String c, int y){

        //Three-Argument Constructor

        if(bal>0 && c.length()==13 && y>0){

            balance = bal;

            cnic = c;

            yearofOpening = y;

        }

    }

    public void setValues(int bal, String c, int y){

        if(bal>0 && c.length()==13 && y > 0){

            balance = bal;

            cnic = c;

            yearofOpening = y;

        }

    }

    public void display(){

        System.out.println("The balance is: " + balance);

        System.out.println("The cnic is: " + cnic);

        System.out.println("The year of opening account is: " + yearofOpening );

    }

    public int deposit(int moneyToDeposit){

        if(moneyToDeposit < 0){

            System.out.println("You cannot enter negative withdrawal amount");

            return balance;

        }

        else{

            balance = moneyToDeposit + balance;

            return balance;

        }

    }

    public int withdraw(int moneyToWithdraw){

        if(moneyToWithdraw < 0){

            System.out.println("You cannot enter negative deposit money ");

            return balance;

        }

        else{

            balance = moneyToWithdraw - balance;

            return balance;

        }

    }

    public int calculateAgeOfAccount(int currentYear){

        return currentYear - yearofOpening;

    }

}

**Activity – 2**

# **Source Code:**

public class Book {

    String author;

    String[] chapterName = new String[5];

    public Book(){

        //default constructor

    }

    public Book(String auth, String[] chpName){

        if((auth.length()==0) && (chpName.length==0)){

            System.out.println("author name & chapter name is empty!!");

        }

        else{

            author = auth;

            chapterName = chpName;

        }

    }

    public void display(){

        System.out.println("author name is: " + author);

        for(int i =0; i < chapterName.length; i++){

            System.out.println(chapterName[i] + " ");

        }

    }

    public void setValues(String auth, String[] chpName){

        author = auth;

        chapterName = chpName;

    }

    public boolean checkIfAuthorNameStartsWithA(){

        if(author.charAt(0) == 'A'){

            return true;

        }

        else{

            return false;

        }

    }

    //boolean return

    public boolean searchChapter(String chpName){

        for(int i =0; i < chapterName.length; i++){

            if(chapterName[i].equals(chpName)){

                return true;

            }

        }

        return false;

    }

}

**Activity – 3**

# **Source Code:**

//circle class

public class Circle{

    double radius;

    public void setValues(double r){

        radius = r;

    }

    public void display(){

        System.out.print("The radius of the circle is: " + radius);

    }

    public void findDiameter(){

        double diameter = 2\*radius;

        System.out.print("The diameter of the circle is: " + diameter);

    }

    public void findCircumfrence(){

        double circumfrence = 2\*(3.14)\*radius;

        System.out.print("The circumfrence of the circle is: " + circumfrence);

    }

    public void findArea(){

        double area = 2\*(3.14)\*(radius\*radius);

        System.out.print("The area of the circle is: " + area);

    }

}

**Activity – 4**

# **Source Code:**

public class Point {

    int x;

    int y;

    public Point(){

        //Default Constructor

    }

    public Point(int a, int b){

        //Two-Argument Constructor

        if((a < 0 && b < 0) || (a > 0 && b > 0)){

            x = a;

            y = b;

        }

    }

    public void display(){

        System.out.println("value of x is: " + x);

        System.out.println("value of y is: " + y);

    }

    public void setValues(int a, int b){

        x = a;

        y = b;

    }

    public void move(int dx, int dy){

        x = x + dx;

        y = y + dy;

        System.out.println("moved value of x & y is: (" + x + ", " + y + ")");

    }

        public boolean checkOrigin(){

            if((x == 0) && (y == 0)){

                return true;

            }

            else{

                return false;

            }

        }

}

**Activity – 5**

# **Source Code:**

public class QuadraticEquation {

    double a;

    double b;

    double c;

    public QuadraticEquation(){

        // Default Constructor

    }

    public QuadraticEquation(double x, double y){

        // Two-Argument Constructor

        a = x;

        b = y;

    }

    public void display(){

        System.out.println("value of a is: " + a);

        System.out.println("value of b is: " + b);

        System.out.println("value of c is: " + c);

    }

    public void setValues(double x, double y, double z){

        if(x < 0.0 && y < 0.0 && z < 0.0){

            System.out.println("Negative values are not allowed");

        }

        else{

            a = x;

            b = y;

            c = z;

        }

    }

    //return double

    public double getDiscriminant(){

        double discriminant = ((b\*b)-4\*(a\*c));

        return discriminant;

    }

    public boolean checkIfDescriminantIsGretaerThan100(){

        double disc = getDiscriminant();

        if(disc > 100.0){

            return true;

        }

        else{

            return false;

        }

    }

}

**Activity – 6**

# **Source Code:**

public class Rectangle {

    int length;

    int width;

    public Rectangle(){

        // Default Constructor

    }

    public Rectangle(int l , int w){

        // Two-Argument Constructor

        length = l;

        width = w;

    }

    public void display(){

        System.out.println(length);

        System.out.println(width);

    }

    public void setValues(int l, int w){

        if(l < 0 && w < 0){

            System.out.println("length & width can't be zero");

        }

        else{

            length = l;

            width = w;

        }

    }

    public int calculateArea(){

        int area = length \* width;

        return area;

    }

    public boolean checkSquare(){

        if(length == width){

            return true;

        }

        else{

            return false;

        }

    }

}

**Activity – 7**

# **Source Code:**

public class Student {

    String name;

    double gpa;

    String[] subjects = new String[5];

    String email;

    public Student(){

        //default constructor

    }

    public Student(String n, double g, String[] subj, String e){

        //Four-Argument constructor

        name = n;

        gpa = g;

        subjects = subj;

        email = e;

    }

    public void setValues(String n, double g, String[] subj, String e){

        name = n;

        gpa = g;

        subjects = subj;

        email = e;

    }

    public boolean searchSubject(String subjectName){

        for(int i = 0; i<subjects.length; i++){

            if(subjectName.equalsIgnoreCase(subjects[i])){

                return true;

            }

        }

            return false;

    }

    public boolean checkProbStatus(){

        if(gpa >= 2.0){

            return false;

        }

        else{

            return true;

        }

    }

    //boolean return

    public boolean validateEmail(){

        if(email.contains("@gmail.com")){

            return true;

        }

        else{

            return false;

        }

    }

}

**Activity – 8**

# **Source Code:**

public class University {

    String uniName;

    String location;

    String rectorName;

    String[] departments = new String[20];

    public University(){

        //no-argument constructor

    }

    public University(String uName, String rectName){

        //two-arguments constructor

        uniName = uName;

        rectorName = rectName;

    }

    public University(String uName, String ulocation, String rectName, String[] dep){

        uniName = uName;

        location = ulocation;

        rectName = rectName;

        if(dep.length <= departments.length){

            for(int i= 0; i < dep.length; i++){

                departments[i] = dep[i];

            }

        }

        else{

            for(int i=0; i<departments.length;i++){

                departments[i] = dep[i];

            }

        }

    }

    //methods

    public void display(){

        System.out.println("uni name is: " + uniName);

        System.out.println("uni location is: " + location);

        System.out.println("uni rector name is: " + rectorName);

        //for printing departments name

        for(int i=0;i<departments.length;i++){

            if(departments[i]!= null){

                System.out.println(departments[i]);

                }

            }

        }

public void setValues(String uName, String ulocation, String rectName, String[] dep){

            uniName = uName;

            location = ulocation;

            rectName = rectName;

            if(dep.length<=departments.length){

                for(int i=0;i<dep.length;i++){

                    departments[i] = dep[i];

                }

            }

            else{

                for(int i=0; i<departments.length;i++){

                    departments[i] = dep[i];

                }

            }

        }

    public void addDepartments(String depName){

        if(departments[departments.length-1] != null){

            System.out.println("The department list is already filled! we cannot add "+ depName);

        }

        else{

            for(int i= 0; i < departments.length; i++){

                if(departments[i]==null){

                    departments[i] = depName;

                    break;

            }

        }

    }

}

    public boolean checkIfLocatedInCapital(){

        if(location.equals("Islamabad")){

            return true;

        }

        else{

            return false;

        }

    }

    public boolean searchDepartment(String dept){

        for(int i =0; i < departments.length; i++){

            if(departments[i].equalsIgnoreCase(dept)){

                return true;

            }

        }

        return false;

    }

}

**Runner Class**

# **Source Code:**

public class RunnerUniversity {

    public static void main(String args[]){

        University u1 = new University();

        String[] departments = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10"};

        u1.setValues("comsats", "islamabad", "ali", departments);

        u1.display();

        u1.addDepartments("11");

        u1.checkIfLocatedInCapital();

        u1.searchDepartment("1");

    }

}

**Runner Class**

# **Source Code:**

public class Runner{

    public static void main(String args[]){

        // // Q1

        // Account a1 = new Account();

        // a1.setValues(1200, "1234567891234", 2022);

        // a1.display();

        // // a1.Account();        //Default Constructor

        // // a1.Account(1500);    //One-Argument Constructor

        // // a1.Account(2400, "3452348976345", 2021);    //Three-Argument Constructor

        // a1.deposit(1000);

        // a1.withdraw(500);

        // a1.calculateAgeOfAccount(2022);

        // Q2

        QuadraticEquation q1 = new QuadraticEquation();

        q1.setValues(2.5, 5.1, 10.6);

        q1.display();

        q1.getDiscriminant();

        q1.checkIfDescriminantIsGretaerThan100();

    }

}