P22CS303

OPP'S WITH JAVA LAB MANUAL

SEMESTER – III

Prof Chaitra B.P, Assistant professor
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDEX

	□ Permit withdrawal and update the balance □ Check for the minimum balance(only for Current account), impose penalty if necessary and update the balance.	
7	Design a base class Circle with member variables (radius of type double and color of type character), methods (getRadius(), getArea()) and constructors (Circle(radius), Circle(radius, color)). Derive subclass called Cylinder from the super class Circle with member variable (height) of type double, public methods (getHeight(), getVolume(), getArea()) and constructors(Cylinder(height),Cylinder(height,radius),Cylinder(height, radius, color)). Create the two instances of cylinder and print similar cylinders if the area, volume and color of cylinders are same. Demonstrate the code reuse and polymorphism properties of Object oriented programming by inheriting the constructors and methods of the base class.	30
8	Create an interface with name encryption and members as message and encrypt(). Derive two classes from this interface namely Nextchar and Prevchar In Nextchar class implement the methodencrypt() to replace each character by its next character. In Prevcharclass implement the method encrypt() to replace each character by its previous character. For example: 1. If the input is "college" then the output is "dpmmfh" (replace each character by next characher). If the input is zebra then the output is "ydaqz" (replace each character by previous character).	34
9	Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class internals has an array that stores the internal marks scored in six courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in six courses of the current semester of the student. Import the two packages in a file that declares the final marks of N students in all six courses.	38
10	Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number	45

11	Write a java program to handle the following exceptions based on choice made by the user by writing suitable try and catch block. i) Arithmetic Exception ii) Array Index Out Of Bounds Exception iii) Number Format Exception iv) String Index Out Of Bound Exception v) Null Pointer Exception	49
12	Define a class Sort with generic method by name Arrange(T[]) and Display(T[]). Write a program to sort array elements of different data types.	57

1. Accept N numbers and find their sum. Check whether the sum is prime or not.

```
package VasanthKumar;
import java.util.*;
public class prime
 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the value of N:");
        int n = in.nextInt();
        int sum = 0;
        System.out.println(" Input "+ n + " Numbers:");
        for (int i = 0; i < n; i++)
       {
            sum += in.nextInt();
        }
        if (sum < 2)
       System.out.println(sum + " is not a prime number.");
        else
        {
            boolean isPrime = true;
            for (int i = 2; i <= sum / 2; i++)
             {
                if (sum % i == 0) {
                    isPrime = false;
                    break;
                }
         }
            if (isPrime)
                System.out.println(sum + " is a prime number.");
            else
                System.out.println(sum + " is not a prime number.");
        }
    }
}
```

Output-1:

```
Enter the value of N:
 4
 Input 4 Numbers:
1
4
6
3
     14 is not a prime number.
```

Output-2:

Enter the value of N:

5

Input 5 Numbers:

1

2

6

8

-4

13 is a prime number.

2. Evaluate the following series using switch statement

a) a + 2a/b + 3a/2b + ... + (n)a/(n-1)b

b) 1 + 1/2 + 1/4 + 1/8+ . . . . . .

```
package VasanthKumar;
import java.util.Scanner;
public class Series {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        while (true) {
            System.out.println("Enter 1 to evaluate series: a + 2a/b + 3a/2b + ... + na/(n-1)b");
            System.out.println("Enter 2 to evaluate series: 1 + 1/2 + 1/4 + 1/8 + ...");
            System.out.println("Enter any other number to exit.");
            int choice = in.nextInt();

            if (choice == 1 || choice == 2) {
                System.out.println("Enter value of n:");
                int n = in.nextInt();
                double sum = 0.0;

                switch (choice) {
                    case 1:
                        System.out.println("Enter value of a and b:");
                        int a = in.nextInt();
                        int b = in.nextInt();
                        sum = a;
                        for (int i = 2; i <= n; i++) {
                            sum += (double) (i * a) / ((i - 1) * b);
                        }
                        break;
                    case 2:
                        for (int i = 0; i < n; i++) {
                            sum += 1.0 / Math.pow(2, i);
                        }
                        break;
                }
                System.out.println("Result: " + sum);
            } else {
                System.out.println("Exiting program.");
                break;
            }
        }
    }

}
```

Output:

```
Enter 1 to evaluate series: a + 2a/b + 3a/2b + ... + na/(n-1)b
Enter 2 to evaluate series: 1 + 1/2 + 1/4 + 1/8 + ...
Enter any other number to exit.
1
Enter value of n:
4
Enter value of a and b:
1
2
Result: 3.4166666666666665

Enter 1 to evaluate series: a + 2a/b + 3a/2b + ... + na/(n-1)b
Enter 2 to evaluate series: 1 + 1/2 + 1/4 + 1/8 + ...
Enter any other number to exit.
2
Enter value of n:
3
Result: 1.75

Enter 1 to evaluate series: a + 2a/b + 3a/2b + ... + na/(n-1)b
Enter 2 to evaluate series: 1 + 1/2 + 1/4 + 1/8 + ...
Enter any other number to exit.
3
Exiting program.
```

3.To read a string and the two index values (i and j). Extract the string from i^{th} position to j^{th} position

```
package VasanthKumar;
import java.util.*;

public class SubString {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String inputString = "";

        System.out.print("Enter the input string: ");
        inputString = scanner.nextLine();

        System.out.print("Enter the starting index (i): ");
        int startIndex = scanner.nextInt();

        System.out.print("Enter the ending index (j): ");
        int endIndex = scanner.nextInt();

        if (startIndex < 0 || endIndex >= inputString.length() || startIndex > endIndex) {
            System.out.println("Invalid index positions.");
            System.exit(0);
        }

        String extractedSubstring = "";
        for (int index = startIndex; index <= endIndex; index++) {
            extractedSubstring += inputString.charAt(index);
        }

        System.out.println("Extracted Substring: " + extractedSubstring);
    }
}
```

Output-1:

```
Enter the input string: Vasanth kumar N P
Enter the starting index (i): 8
Enter the ending index (j): 15
    Extracted Substring: kumar N.
```

Output-2:

```
Enter the input string: Bharath
Enter the starting index (i): 2
Enter the ending index (j): 9
        Invalid index positions.
```

Output-3:

```
Enter the input string: Jeevan
Enter the starting index (i): -1
Enter the ending index (j): 3
    Invalid index positions.
```

4).Create a Java class called Complex with the following details and variables within it as (i) Real (ii) Imaginary
Develop a Java program to perform addition and subtraction of two complex numbers by using the method add() and subtract( ) respectively by passing object as parameter and display result using method display(). Initialize the real and imaginary values of the complex number using parameterized constructor.

```
package vasanthKumar;
public class Complex {
    private float real;
    private float imaginary;

    public Complex(float real, float imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }

    public Complex add(Complex obj) {
        float sumReal = this.real + obj.real;
        float sumImaginary = this.imaginary + obj.imaginary;
        return new Complex(sumReal, sumImaginary);
    }
    public Complex subtract(Complex obj) {
        float diffReal = this.real - obj.real;
        float diffImaginary = this.imaginary - obj.imaginary;
        return new Complex(diffReal, diffImaginary);
    }

    public void display() {
        System.out.printf("%.1f%s%.1fi%n", real, (imaginary >= 0 ? "+" : ""), imaginary);
    }

    public static void main(String[] args) {
        Complex C1 = new Complex(-5, -4);
        Complex C2 = new Complex(-6, -6);


        Complex resultAddition = C1.add(C2);
        System.out.print("Sum of two complex numbers = ");
        resultAddition.display();


        Complex resultSubtraction = C1.subtract(C2);
        System.out.print("Difference of two complex numbers = ");
        resultSubtraction.display();
    }
}
```

Output:

```
Sum of two complex numbers = -11.0-10.0i
Difference of two complex numbers = 1.0+2.0i
```

5).Define a class named MyTime to represent a time instance with private instance variables: hour (ranging from 0 to 23) and minute (ranging from 0 to 59). The constructor should call the setTime() method to set the instance variables. The setTime() method should check if the provided hour and minute values are valid before setting the instance variables. Additionally, define methods getHour(), getMinute(), nextMinute(), and nextHour(). The nextMinute() method should update the instance to the next minute and return this instance. Note that the nextMinute() of 23:59 is 00:00. The nextHour() method should behave similarly. Write the code for the MyTime class and also create a test program named TestMyTime to test all the methods defined in the MyTime class. Avoid providing a program directly.

```
package vasanthKumar;
class MyTime {
    private int hour;
    private int minute;

    MyTime(int hour, int minute) {
        setTime(hour, minute);
    }

    public void setTime(int hour, int minute) {
        if (hour >= 0 && hour <= 23 && minute >= 0 && minute <= 59) {
            this.hour = hour;
            this.minute = minute;
        } else {
            System.out.println("Invalid time. Hour should be between 0 and 23, and minute between 0 and 59.");
        }
    }

    public int getHour() {
        return hour;
    }

    public int getMinute() {
        return minute;
    }

    public MyTime nextMinute() {
        minute++;
        if (minute >= 60) {
            minute = 0;
            hour = (hour + 1) % 24;
        }
        return this;
    }

    public MyTime nextHour() {
        hour = (hour + 1) % 24;
        return this;
    }
}

class TestMyTime {
```

```
    public static void main(String[] args) {
        MyTime t = new MyTime(23, 59);

        System.out.println("Current Time: " + t.getHour() + ":" + t.getMinute());

        t.nextMinute();
        System.out.println("Next Minute: " + t.getHour() + ":" + t.getMinute());

        t.nextHour();
        System.out.println("Next Hour: " + t.getHour() + ":" + t.getMinute());
    }
}
```

Output:

```
Current Time: 23:59
Next Minute: 0:0
Next Hour: 1:0
```

6).Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their requirements. The savings account provides compound interest and withdrawal facilities. The current account does not provide interest. Current account holders should also maintain a minimum balance (Rs 5000) and if the balance falls below this level, a service charge (Rs 100) is imposed. Include the necessary methods in order to achieve the following tasks:

- **Accept deposit from customer and update the balance.**
- **Display the balance.**
- **Compute and deposit interest**
- **Permit withdrawal and update the balance**
- **Check for the minimum balance (only for Current account), impose penalty if necessary and update the balance.**

```
package VasanthKumar;
import java.util.Scanner;

class Account {
    String customerName;
    long accountNumber;
    String accountType;
    double balance=5000.0;

    void customerDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Customer Name: ");
        customerName = sc.nextLine();
        System.out.print("Enter Account Number: ");
        accountNumber = sc.nextLong();
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposition of Rs: " + amount + " is successful ");
    }

    void displayDetails() {
        System.out.println("Customer Name: " + this.customerName);
        System.out.println("Account Number: " + this.accountNumber);
    }

    void displayBalance() {
        System.out.println("Your Account Balance Is Rs: " + this.balance);
    }
}

class CurrAcct extends Account {
```

```
    double minBalance = 5000;
    double serviceCharge = 100;

    CurrAcct(String accountType) {
        this.accountType = accountType;
    }

    void withdraw(double amount) {
        if (amount <= (this.balance - minBalance)) {
            this.balance -= amount;
            System.out.println("Withdrawal of rupees " + amount + " is successful.");
                }
        else {
            System.out.println("Insufficient balance. Withdrawal unsuccessful.");
            System.out.println("Service charge of Rs " + serviceCharge + " has imposed. So your Updated
Balance is Rs " + this.balance);
        }
    }
}

class SavAcct extends Account {
    double interestRate = 0.03;
    double interest=0.0;
    SavAcct(String accountType) {
        this.accountType = accountType;
    }
    void compute_interest(double amt) {
        interest += (amt * interestRate);
        balance += interest;
    }
    void interest() {
        System.out.println("Interest of Rs " + interest + " developed.");
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal of rupees " + amount + " is successful.");
        } else {
            System.out.println("Insufficient balance. Withdrawal unsuccessful.");
        }
    }
}

class Mainm {
    public static void main(String[] args) {
```

```
        int opt1, opt2, choice;
        Scanner sc = new Scanner(System.in);

        CurrAcct currentAccount = new CurrAcct("Current");
        SavAcct savingsAccount = new SavAcct("Savings");

        do {
            System.out.println("Enter 1: to Open Current Account\nEnter 2: to Open Saving
Account\nEnter 3: to Terminate");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    do {
                        System.out.println("\nEnter 1: to Enter Customer Details\nEnter 2: to Deposit
Money\nEnter 3: to Display Balance\nEnter 4: to Withdraw\nEnter 5: to Display Customer
details\nEnter 6: to Go back to menu");
                        opt1 = sc.nextInt();
                        switch (opt1) {
                            case 1:
                                currentAccount.customerDetails();
                                break;
                            case 2:
                                double B;
                                do {
                                    System.out.println("Enter the Amount to Deposit to the Current Account: ");
                                    B = sc.nextDouble();
                                }while(B<0);
                                currentAccount.deposit(B);
                                break;
                            case 3:
                                currentAccount.displayBalance();
                                break;
                            case 4:
                                System.out.println("Enter the Amount to withdraw: ");
                                double E = sc.nextDouble();
                                currentAccount.withdraw(E);
                                break;
                            case 5:
                                currentAccount.displayDetails();
                                break;
                        }
                    } while (opt1 != 6);
                    break;
                case 2:
                    do {
```

```
                System.out.println("\nEnter 1: to Enter Customer Details\nEnter 2: to Deposit
Money\nEnter 3: to Check Interest developed\nEnter 4: to Display Balance\nEnter 5: to Withdraw
Money\nEnter 6: to Display Customer details\nEnter 7: to Go back to menu");
                opt2 = sc.nextInt();
                switch (opt2) {
                    case 1:
                        savingsAccount.customerDetails();
                        break;
                    case 2:
                        double C;
                        do {
                            System.out.println("Enter the Amount to Deposit to the Saving Account: ");
                            C = sc.nextDouble();
                        }while(C<0);
                        savingsAccount.deposit(C);
                        savingsAccount.compute_interest( C);
                        break;
                    case 3:
                        savingsAccount.interest();
                        break;
                    case 4:
                        savingsAccount.displayBalance();
                        break;
                    case 5:
                        System.out.println("Enter the Amount to withdraw: ");
                        double A = sc.nextDouble();
                        savingsAccount.withdraw(A);
                        break;
                    case 6:
                        savingsAccount.displayDetails();
                        break;
                }
            } while (opt2 != 7);
            break;
        case 3:
            System.exit(0);
        }
    } while (choice != 3);
  }
}
```

Output:

```
Enter 1: to Open Current Account
Enter 2: to Open Saving Account
Enter 3: to Terminate
1

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Display Balance
Enter 4: to Withdraw
Enter 5: to Display Customer details
Enter 6: to Go back to menu
1
Enter Customer Name: Bharath
Enter Account Number: 100

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Display Balance
Enter 4: to Withdraw
Enter 5: to Display Customer details
Enter 6: to Go back to menu
2
Enter the Amount to Deposit to the Current Account:
2500
Deposition of Rs: 2500.0 is successful

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Display Balance
Enter 4: to Withdraw
Enter 5: to Display Customer details
Enter 6: to Go back to menu
3
Your Account Balance Is Rs: 7500.0

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Display Balance
Enter 4: to Withdraw
Enter 5: to Display Customer details
Enter 6: to Go back to menu
4
Enter the Amount to withdraw:
3000
```

Insufficient balance. Withdrawal unsuccessful.
Service charge of Rs 100.0 has imposed. So your Updated Balance is Rs 7500.0

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Display Balance
Enter 4: to Withdraw
Enter 5: to Display Customer details
Enter 6: to Go back to menu
5
Customer Name: Bharath
Account Number: 100

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Display Balance
Enter 4: to Withdraw
Enter 5: to Display Customer details
Enter 6: to Go back to menu
6
Enter 1: to Open Current Account
Enter 2: to Open Saving Account
Enter 3: to Terminate
2

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Check Interest developed
Enter 4: to Display Balance
Enter 5: to Withdraw Money
Enter 6: to Display Customer details
Enter 7: to Go back to menu
1
Enter Customer Name: Sudeep
Enter Account Number: 123

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Check Interest developed
Enter 4: to Display Balance
Enter 5: to Withdraw Money
Enter 6: to Display Customer details
Enter 7: to Go back to menu
2
Enter the Amount to Deposit to the Saving Account:
1000

Deposition of Rs: 1000.0 is successful

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Check Interest developed
Enter 4: to Display Balance
Enter 5: to Withdraw Money
Enter 6: to Display Customer details
Enter 7: to Go back to menu
3
Interest of Rs 30.0 developed.

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Check Interest developed
Enter 4: to Display Balance
Enter 5: to Withdraw Money
Enter 6: to Display Customer details
Enter 7: to Go back to menu
4
Your Account Balance Is Rs: 6030.0

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Check Interest developed
Enter 4: to Display Balance
Enter 5: to Withdraw Money
Enter 6: to Display Customer details
Enter 7: to Go back to menu
5
Enter the Amount to withdraw:
10000
Insufficient balance. Withdrawal unsuccessful.

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Check Interest developed
Enter 4: to Display Balance
Enter 5: to Withdraw Money
Enter 6: to Display Customer details
Enter 7: to Go back to menu
5
Enter the Amount to withdraw:
100
Withdrawal of rupees 100.0 is successful.

```
Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Check Interest developed
Enter 4: to Display Balance
Enter 5: to Withdraw Money
Enter 6: to Display Customer details
Enter 7: to Go back to menu

6
Customer Name: Sudeep
Account Number: 123

Enter 1: to Enter Customer Details
Enter 2: to Deposit Money
Enter 3: to Check Interest developed
Enter 4: to Display Balance
Enter 5: to Withdraw Money
Enter 6: to Display Customer details
Enter 7: to Go back to menu
7
Enter 1: to Open Current Account
Enter 2: to Open Saving Account
Enter 3: to Terminate
3
```

Alternative Program:

```
import java.util.Scanner;

class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance = 5000;

    Scanner scanner = new Scanner(System.in);

    void read(String str) {
        this.accountType = str;
        System.out.println("Enter Customer Name: ");
        this.customerName = scanner.nextLine();
        System.out.println("Enter Account Number: ");
        this.accountNumber = scanner.nextInt();
```

```
    }

    void deposit() {
        System.out.println("Enter amount to deposit: ");
        double amount = scanner.nextDouble();
        this.balance += amount;
        System.out.println("Amount deposited successfully.");
    }

    void display() {
        System.out.println("Account Type: " + this.accountType);
        System.out.println("Customer Name: " + this.customerName);
        System.out.println("Account Number: " + this.accountNumber);
        System.out.println("Account Balance: " + this.balance);
    }
}

class SavingsAccount extends Account {
    double interestRate = 0.03;

    void withdraw() {
        System.out.println("Enter amount to withdraw: ");
        double amount = scanner.nextDouble();
        if (this.balance >= amount) {
            this.balance -= amount;
            System.out.println("Amount withdrawn successfully.");
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    void computeInterest() {
        double interest = this.balance * interestRate;
        this.balance += interest;
        System.out.println("Interest of Rs. " + interest + " has been Developed.");
    }
}

class CurrentAccount extends Account {
    void withdraw() {
        System.out.println("Enter amount to withdraw: ");
        double amount = scanner.nextDouble();
        int check = checkMin(amount);
        if (check == 1 && this.balance >= amount) {
            this.balance -= amount;
            System.out.println("Amount withdrawn successfully.");
        }
    }

    int checkMin(double amount) {
```

```
        if ((this.balance - amount) >= 5000.0) {
            return 1;
        } else {
            System.out.println("Service charge of Rs. 100 has been imposed.");
            this.balance -= 100;
            return 0;
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        CurrentAccount CA = new CurrentAccount();
        SavingsAccount SA = new SavingsAccount();

        System.out.println("Choose the type of account : \n 1: Current \n2: Saving \n");
        int ch = in.nextInt();

        switch (ch) {
            case 1:
                CA.read("current Account");
                CA.deposit();
                CA.withdraw();
                CA.display();
                break;
            case 2:
                SA.read("Savings Account");
                SA.deposit();
                SA.computeInterest();
                SA.withdraw();
                SA.display();
                break;
            case 3:
                System.exit(0);
        }
    }
}
```

7).Design a base class Circle with member variables (radius of type double and color of type character), methods (getRadius(), getArea()) and constructors (Circle(radius), Circle(radius, color)). Derive subclass called Cylinder from the super class Circle with member variable (height) of type double, public methods (getHeight(), getVolume(), getArea()) and constructors(Cylinder(height), Cylinder(height,radius), Cylinder(height, radius, color)). Create the two instances of cylinder and print similar cylinders if the area, volume and color of cylinders are same. Demonstrate the code reuse and polymorphism properties of Object-oriented programming by inheriting the constructors and methods of the base class.

```
package VasanthKumar;

import java.util.Scanner;

class Circle {
    double radius;
    char color;

    public Circle(double radius) {
        this.radius = radius;
    }

    public Circle(double radius, char color) {
        this.radius = radius;
        this.color = color;
    }

    public double getRadius() {
        return radius;
    }

    public double getArea() {
        return Math.PI * radius * radius;
    }
}

class Cylinder extends Circle {
    double height;

    public Cylinder(double height) {
        super(1.0);
        this.height = height;
    }

    public Cylinder(double height, double radius) {
        this(height);
        this.radius = radius;
    }

    public Cylinder(double height, double radius, char color) {
        this(height, radius);
```

```
        this.color = color;
    }

    public double getHeight() {
        return height;
    }

    public double getVolume() {
        return getArea() * getHeight();
    }

    public double getArea() {
        return 2 * super.getArea() + 2 * Math.PI * getRadius() * getHeight();
    }
}

 class Shape {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the radius of cylinder 1:");
        double radius1 = scanner.nextDouble();
        System.out.println("Enter the height of cylinder 1:");
        double height1 = scanner.nextDouble();
        System.out.println("Enter the color of cylinder 1:");
        char color1 = scanner.next().charAt(0);

        System.out.println("Enter the radius of cylinder 2:");
        double radius2 = scanner.nextDouble();
        System.out.println("Enter the height of cylinder 2:");
        double height2 = scanner.nextDouble();
        System.out.println("Enter the color of cylinder 2:");
        char color2 = scanner.next().charAt(0);

        Cylinder cylinder1 = new Cylinder(height1, radius1, color1);
        Cylinder cylinder2 = new Cylinder(height2, radius2, color2);

        if (cylinder1.getArea() == cylinder2.getArea() && cylinder1.getVolume() == cylinder2.getVolume() &&
cylinder1.color == cylinder2.color) {
            System.out.println("The cylinders are similar.");
        } else {
            System.out.println("The cylinders are not similar.");
        }
    }
}
```

Output-1:

```
Enter the radius of cylinder 1:
2.5
Enter the height of cylinder 1:
8.8
Enter the color of cylinder 1:
G
Enter the radius of cylinder 2:
2.5
Enter the height of cylinder 2:
8.8
Enter the color of cylinder 2:
G
The cylinders are similar.
```

Output-2:

```
Enter the radius of cylinder 1:
1.2
Enter the height of cylinder 1:
2
Enter the color of cylinder 1:
y
Enter the radius of cylinder 2:
3.2
Enter the height of cylinder 2:
2
Enter the color of cylinder 2:
y
The cylinders are not similar.
```

8). Create an interface with name encryption and members as message and encrypt(). Derive two classes from this interface namely Nextchar and Prevchar In Nextchar class implement the method encrypt() to replace each character by its next character. In Prevchar class implement the method encrypt() to replace each character by its previous character.
For example: 1. If the input is "college" then the output is "dpmmfh" (replace each character by next characher). If the input is zebra then the output is "ydaqz" (replace each character by previous character).

```
package VasanthKumar;
import java.util.*;

interface Encryption {
    String message = "";
    void encrypt();
}

class NextChar implements Encryption {
    String message = "";

    NextChar(String message) {
        this.message = message;
    }

    public void encrypt() {
        String result = "";
        int i = 0, j = message.length();
        while (i < j) {
            char c = message.charAt(i);
            if (Character.isLetter(c)) {
                if (c == 'z') c = 'a';
                else if (c == 'Z') c = 'A';
                else c++;
            } else if (Character.isDigit(c)) {
                if (c == '9') c = '0';
                else c++;
            }
            result += c;
            i++;
        }
        System.out.println("Encrypted Message= " + result);
    }
}

class PrevChar implements Encryption {
    String message = "";
```

```
    PrevChar(String message) {
        this.message = message;
    }

    public void encrypt() {
        String result = "";
        int i = 0, j = message.length();
        while (i < j) {
            char c = message.charAt(i);
            if (Character.isLetter(c)) {
                if (c == 'a') c = 'z';
                else if (c == 'A') c = 'Z';
                else c--;
            } else if (Character.isDigit(c)) {
                if (c == '0') c = '9';
                else c--;
            }
            result += c;
            i++;
        }
        System.out.println("Encrypted Message= " + result);
    }
}

class CharEncrypt {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the message to encrypt next character: ");
        String message1 = in.nextLine();
        NextChar n = new NextChar(message1);
        n.encrypt();
        System.out.println("Enter the message to encrypt previous character: ");
        String message2 = in.nextLine();
        PrevChar p = new PrevChar(message2);
        p.encrypt();
    }
}
```

Output-1:

Enter the message to encrypt next character:
Vasanth Kumar fifth semester D section
Encrypted Message= Wbtboui Lvnbs gjgui tfnftufs E tfdujpo

Enter the message to encrypt previous character:
Java Oriented programming
Encrypted Message= Izuz Nqhdmsdc oqnfqzllhmf

Output-2:

Enter the message to encrypt next character:
Java 2023 batch
Encrypted Message= Kbwb 3134 cbudi

Enter the message to encrypt previous character:
0123456789
Encrypted Message= 9012345678

9). Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class internals has an array that stores the internal marks scored in six courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in six courses of the current semester of the student. Import the two packages in a file that declares the final marks of N students in all six courses.

```
//1st File Name : Student.javaInternals.java   &  package CIE
package CIE;
import java.util.*;
public class Student {
    public Object readD;
    String USN;
    String Name;
    int Sem;
    Scanner in=new Scanner(System.in);
    public void readD()
    {
        System.out.println("Enter the NAME , USN and Sem of the student: ");
        this.Name=in.nextLine();
        this.USN=in.nextLine();
        this.Sem=in.nextInt();
    }
   public  void dispalyD()
   {
       System.out.println("Name : "+this.Name);
       System.out.println("USN : "+this.USN);
       System.out.println("Sem : "+this.Sem);
   }
}
//2nd File Name : Internals.java   &  package CIE

package CIE;
import java.util.Scanner;
 public class Internals
 {
    Scanner in = new Scanner(System.in);
   public int[] C = new int[20];
    public void readC()
    {
        for (int i = 0; i < 6; i++)
        {
            System.out.println("Enter the CIE marks obtained by the student in Course -: " + (i + 1));
            this.C[i] = in.nextInt();
        }
```

```
    }
}

//3rd File Name : External.java   &  package SEE



package SEE;
import CIE.*;

import java.util.Scanner;

public class External extends Student {

        Scanner in = new Scanner(System.in);
      public  int[] S = new int[20];

        public void readS() {
            for (int i = 0; i < 6; i++) {
                System.out.println("Enter the SEE marks obtained by the student in Course - " + (i + 1)+":");
                this.S[i] = in.nextInt();
            }
        }

}

//4th File Name : TotalResult.java   &  package VasanthKumar

Package VasanthKumar;

import CIE.*;
import SEE.*;
import java.util.Scanner;

public class TotalResult {
    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);
        System.out.println("Enter The Count of students: ");
        int N =in.nextInt();
        Student[] s = new Student[N];
        Internals[] i = new Internals[N];
        External[] e = new External[N];
        int[] Total = new int[N];
        int[][] Marks = new int[N][6];
```

```
        for (int j = 0; j < N; j++) {
            s[j] = new Student();
            s[j].readD();
            i[j] = new Internals();
            i[j].readC();
            e[j] = new External();
            e[j].readS();
            Total[j] = 0;
            for (int k = 0; k < 6; k++) {
                Marks[j][k] = (i[j].C[k] + (e[j].S[k]) / 2);
                Total[j] += Marks[j][k];
            }
        }

        for (int j = 0; j < N; j++) {
            System.out.println("\n---------------------------- ");
            System.out.println("Details of Student " + (j + 1));
            s[j].dispalyD();
            System.out.println("final marks Obtained in  :" );
            for(int m=0;m<6;m++)
            System.out.println("Course-"+(m+1) +" = " +Marks[j][m]);
            System.out.println("---------------------------- ");
            System.out.println("Total Marks = " + Total[j]);
            System.out.println("---------------------------- ");
        }
    }
}
```

Output:

```
Enter The Count of students:
2
Enter the NAME , USN and Sem of the student:
Vasanth Kumar
4PS22CS188
3
Enter the CIE marks obtained by the student in Course -: 1
45
Enter the CIE marks obtained by the student in Course -: 2
46
Enter the CIE marks obtained by the student in Course -: 3
47
Enter the CIE marks obtained by the student in Course -: 4
43
Enter the CIE marks obtained by the student in Course -: 5
49
Enter the CIE marks obtained by the student in Course -: 6
50
Enter the SEE marks obtained by the student in Course - 1:
100
Enter the SEE marks obtained by the student in Course - 2:
99
Enter the SEE marks obtained by the student in Course - 3:
95
Enter the SEE marks obtained by the student in Course - 4:
98
Enter the SEE marks obtained by the student in Course - 5:
90
Enter the SEE marks obtained by the student in Course - 6:
89



Enter the NAME , USN and Sem of the student:
Sudeep
050
5
Enter the CIE marks obtained by the student in Course -: 1
40
Enter the CIE marks obtained by the student in Course -: 2
40
Enter the CIE marks obtained by the student in Course -: 3
10
```

Enter the CIE marks obtained by the student in Course -: 4
20
Enter the CIE marks obtained by the student in Course -: 5
34
Enter the CIE marks obtained by the student in Course -: 6
12
Enter the SEE marks obtained by the student in Course - 1:
50
Enter the SEE marks obtained by the student in Course - 2:
35
Enter the SEE marks obtained by the student in Course - 3:
45
Enter the SEE marks obtained by the student in Course - 4:
56
Enter the SEE marks obtained by the student in Course - 5:
87
Enter the SEE marks obtained by the student in Course - 6:
99

Details of Student 1
Name : Vasanth Kumar
USN : 4PS22CS188
Sem : 3
final marks Obtained in :
Course-1 = 95
Course-2 = 95
Course-3 = 94
Course-4 = 92
Course-5 = 94
Course-6 = 94

Total Marks = 564

Details of Student 2
Name : Sudeep
USN : 050
Sem : 5
final marks Obtained in :
Course-1 = 65

Course-2 = 57
Course-3 = 32
Course-4 = 48
Course-5 = 77
Course-6 = 61

Total Marks = 340
---

10). Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

```
package VasanthKumar;
import java.util.Random;

class Thread1 extends Thread {
    public void run() {
for(int i=0;i<10;i++){
    Random r = new Random();
    int n = r.nextInt(70);
   if (n % 2 == 0) {
        synchronized (Thread2.class) {
            Thread2.number = n;
            Thread2.class.notify();
        }
    } else {
        synchronized (Thread3.class) {
            Thread3.number = n;
            Thread3.class.notify();
        }
    }
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
    }
}
    }
}

class Thread2 extends Thread {
    static int number;

    public void run() {
while(true) {
    synchronized (Thread2.class) {
        try {
            Thread2.class.wait();
            System.out.println("Square of even number " + (number) + " is: " + (number * number));
        } catch (InterruptedException e) {
        }
    }
}
    }
```

```
}

class Thread3 extends Thread {
    static int number;

    public void run() {
      while (true) {
          synchronized (Thread3.class) {
              try {
                  Thread3.class.wait();
                  System.out.println("Cube of odd number  " + (number) + " is: " + (number * number *
number));
              } catch (InterruptedException e) {
              }
          }
      }
    }
}

public class Main {
    public static void main(String[] args) throws InterruptedException {
        Thread1 t1 = new Thread1();
        Thread2 t2 = new Thread2();
        Thread3 t3 = new Thread3();
        t1.start();
        t2.start();
        t3.start();
    }
}
```

Output-1:

Cube of odd number 51 is: 132651
Cube of odd number 13 is: 2197
Square of even number 54 is: 2916
Cube of odd number 41 is: 68921
Cube of odd number 25 is: 15625
Square of even number 0 is: 0
Cube of odd number 27 is: 19683
Square of even number 64 is: 4096
Cube of odd number 1 is: 1
Square of even number 54 is: 2916

Output-2:

Square of even number 8 is: 64
Cube of odd number 39 is: 59319
Cube of odd number 39 is: 59319
Square of even number 12 is: 144
Square of even number 58 is: 3364
Square of even number 12 is: 144
Square of even number 64 is: 4096
Square of even number 4 is: 16
Square of even number 60 is: 3600
Cube of odd number 45 is: 91125

11). Write a java program to handle the following exceptions based on choice made by the user by writing suitable try and catch block.

i) ArithmeticException
ii) ArrayIndexOutOfBoundsException
iii) NumberFormatException
iv) StringIndexOutOfBoundException
v) NullPointerException

```
package VasanthKumar;
import java.util.*;

class ExceptionHandling {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("Choose an exception to check:\n1. ArithmeticException\n2.
ArrayIndexOutOfBoundsException\n3. NumberFormatException\n4.
StringIndexOutOfBoundsException\n5. NullPointerException\n6. EXIT");

            int choice = scanner.nextInt();

            try {
                switch (choice) {
                    case 1:
                        System.out.println("Enter two numbers (numerator and denominator):");
                        int numerator = scanner.nextInt();
                        int denominator = scanner.nextInt();
                        int result = numerator / denominator;
                        System.out.println("Result: " + result);
                        break;
                    case 2:
                        System.out.println("Enter the size of the array:");
                        int arraySize = scanner.nextInt();
                        int[] array = new int[arraySize];
                        System.out.println("Enter the index to access:");
                        int index = scanner.nextInt();
                        int value = array[index];
                        System.out.println("Value: " + value);
                        break;
                    case 3:
                        System.out.println("Enter a string representing an integer:");
                        String integerString = scanner.next();
                        int number = Integer.parseInt(integerString);
```

```
                System.out.println("Parsed Integer: " + number);
                break;
            case 4:
                System.out.println("Enter a string:");
                String inputString = scanner.next();
                System.out.println("Enter the index to access:");
                int stringIndex = scanner.nextInt();
                char character = inputString.charAt(stringIndex);
                System.out.println("Output: " + character);
                break;
            case 5:
                String nullString = null;
                System.out.println(nullString.length());
                break;
            case 6:
                System.out.println("Exiting program.");
                System.exit(0);
            default:
                System.out.println("Invalid choice.");
        }

      } catch (ArithmeticException e) {
        System.out.println("ArithmeticException: " + e.getMessage());
      } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("ArrayIndexOutOfBoundsException: " + e.getMessage());
      } catch (NumberFormatException e) {
        System.out.println("NumberFormatException: " + e.getMessage());
      } catch (StringIndexOutOfBoundsException e) {
        System.out.println("StringIndexOutOfBoundsException: " + e.getMessage());
      } catch (NullPointerException e) {
        System.out.println("NullPointerException: " + e.getMessage());
      }
    }
  }
}
```

Output :

```
Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
1
Enter two numbers (numerator and denominator):
12
2
Result: 6
Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
1
Enter two numbers (numerator and denominator):
1
0
ArithmeticException: / by zero


Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
2
Enter the size of the array:
3
Enter the index to access:
2
Value: 0
Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
```

6. EXIT
2
Enter the size of the array:
2
Enter the index to access:
3
ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 2

Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
3
Enter a string representing an integer:
12
Parsed Integer: 12
Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
3
Enter a string representing an integer:
1.2
NumberFormatException: For input string: "1.2"

Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
4
Enter a string:
vasanth
Enter the index to access:
4
Output: n

Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
4
Enter a string:
vasanth
Enter the index to access:
11
StringIndexOutOfBoundsException: Index 11 out of bounds for length 7

Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
5
NullPointerException: Cannot invoke "String.length()" because "nullString" is null

Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
7
Invalid choice.

Choose an exception to check:
1. ArithmeticException
2. ArrayIndexOutOfBoundsException
3. NumberFormatException
4. StringIndexOutOfBoundsException
5. NullPointerException
6. EXIT
6
Exiting program

12). Define a class Sort with generic method by name Arrange(T[]) and Display(T[]). Write a program to sort array elements of different data types.

```
package vasanth;
import java.util.*;

class Sort<T>
{
    public void Arrange(T[] arr)
    {
        Arrays.sort(arr);
    }

    public void Display(T[] arr) {
        System.out.println("Data in Array:");
        for (T element : arr) {
            System.out.println(element);
        }
    }
}

class SortArr
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        int choice;
        do
        {
            System.out.println("\nMenu:");
            System.out.println("1. Sort String Array");
            System.out.println("2. Sort Integer Array");
            System.out.println("3. Sort Double Array");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            switch (choice) {

                case 1:
                    System.out.println("Enter the size of the string array:");
                    int n1 = scanner.nextInt();
                    String[] SA = new String[n1];
                    System.out.println("Enter " + n1 + " strings:");
                    for (int i = 0; i < n1; i++) {
                        SA[i] = scanner.next();
```

```
            }
            Sort<String> s = new Sort<>();
            s.Arrange(SA);
            s.Display(SA);
            break;

        case 2:
            System.out.println("Enter the size of the integer array:");
            int n2 = scanner.nextInt();
            Integer[] IA = new Integer[n2];
            System.out.println("Enter " + n2 + " integers:");
            for (int i = 0; i < n2; i++) {
                IA[i] = scanner.nextInt();
            }
            Sort<Integer> is = new Sort<>();
            is.Arrange(IA);
            is.Display(IA);
            break;

        case 3:
            System.out.println("Enter the size of the double array:");
            int n3 = scanner.nextInt();
            Double[] DA = new Double[n3];
            System.out.println("Enter " + n3 + " doubles:");
            for (int i = 0; i < n3; i++) {
                DA[i] = scanner.nextDouble();
            }
            Sort<Double> d = new Sort<>();
            d.Arrange(DA);
            d.Display(DA);
            break;
      }

    }while(choice !=4);
  }
}
```

Output:

```
Menu:
1. Sort String Array
2. Sort Integer Array
3. Sort Double Array
4. Exit
Enter your choice: 1
Enter the size of the string array:
4
Enter 4 strings:
vasanth
bharath
jeevan
sudeep

Data in Array:
bharath
jeevan
sudeep
vasanth


Menu:
1. Sort String Array
2. Sort Integer Array
3. Sort Double Array
4. Exit
Enter your choice: 2
Enter the size of the integer array:
5
Enter 5 integers:
91
23
-65
342
76
Data in Array:
-65
23
76
91
342

Menu:
1. Sort String Array
2. Sort Integer Array
```

3. Sort Double Array
4. Exit
Enter your choice: 3
Enter the size of the double array:
3
Enter 3 doubles:
1.2
4.3
0.2
Data in Array:
0.2
1.2
4.3

Menu:
1. Sort String Array
2. Sort Integer Array
3. Sort Double Array
4. Exit
Enter your choice: 4

----------ALL THE BEST----------