B.M.S. College of Engineering

(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science and Engineering

LAB

OBJECT ORIENTED JAVA PROGRAMMING REPORT

23CS3PCOOJ

(December 2023-March 2024)

Submitted by:

Yashraj Sinha

1BM22CS335

B.M.S. College of Engineering
Department of Computer Science and Engineering

Laboratory Certificate

This is to certify that **Yashraj Sinha** has satisfactorily completed the course of Experiments in Practical OBJECT ORIENTED JAVA PROGRAMMING prescribed by the Department during the odd semester 2023-24.

Name of the Candidate: **Yashraj Sinha**

USN No. : **1BM22CS335** Semester: **III** Section: F

Marks	
Max. Marks	Obtained
10	
Marks in Words	

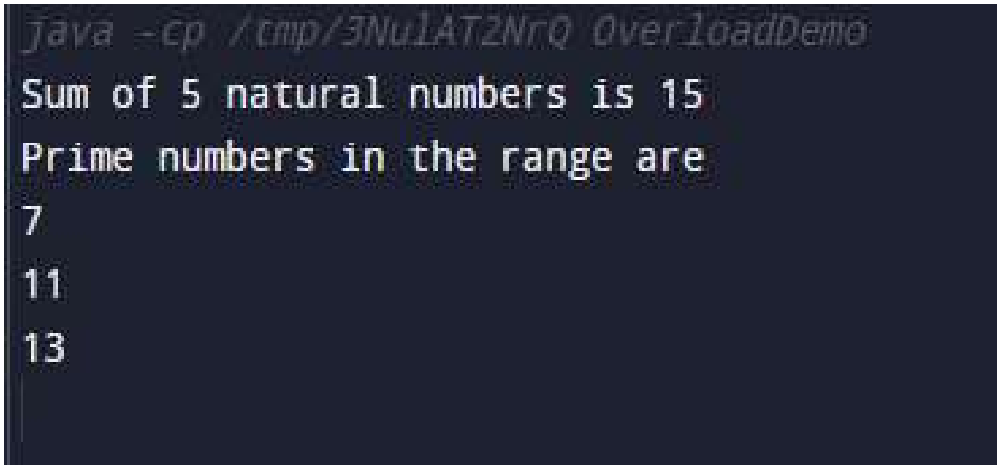
Signature of the staff in-charge **Head of the Department**

Date:

1. Write a program to overload the method print that prints sum of n natural numbers when one variable is passed, and prints the prime numbers in a given range when 2 parameters are passed.

code:

```
class Overload {
void print(int n) {
int sum = 0;
for(int i=1; i<=n;i++) {
sum = sum+i;
}
System.out.println("Sum of "+n+" natural numbers is "+sum);
}
void print(int m, int n) {
System.out.println("Prime numbers in the range are ");
for(int i=m;i<=n;i++) {
int flag=0;
for(int j=2;j<=i/2;j++) {
if(i%j == 0) {
flag = 1;
break;
}
}=0)
System.out.println(i);
}
}
}
class OverloadDemo {
public static void main(String[] args) {
Overload o = new Overload();
o.print(5);
o.print(7,13);
}
}
```

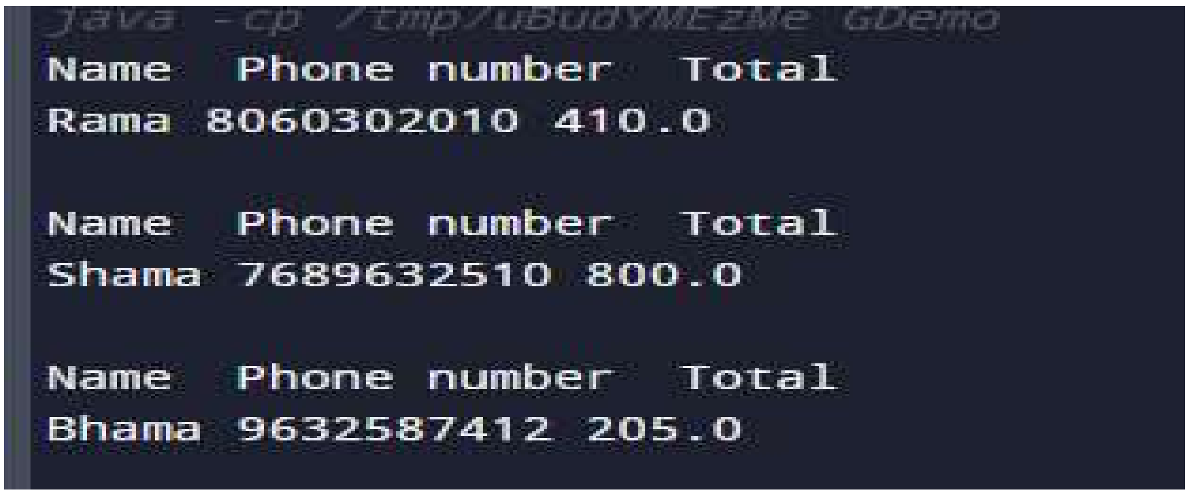


```
java -cp /tmp/3NulAT2NrQ OverloadDemo
Sum of 5 natural numbers is 15
Prime numbers in the range are
7
11
13
```

2. Write a Java program to create a class Grocery that has the variables c_name and c_phone. Create a method to accept 3 parameters to specify quantity of dal, quantity of pulses an quantity of sugar. The method to return the total price. Display the name, ph_no and total bill of 3 customers.

code:

```
class Grocery {
String c_name;
String c_ph;
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_sugar*50;
}
void display()
{
System.out.println("Name "+" "+"Phone number "+" "+"Total");
System.out.println(c_name+" "+c_ph+" "+total);
System.out.println();
}
}
class GDemo {
public static void main(String[] args) {
Grocery g1 = new Grocery("Rama","8060302010");
Grocery g2 = new Grocery("Shama","7689632510");
Grocery g3 = new Grocery("Bhama", "9632587412");
g1.calc(2, 2, 1);
g1.display();
g2.calc(3, 5, 2);
g2.display();
g3.calc(1, 1, 0.5);
g3.display();
}
}
```



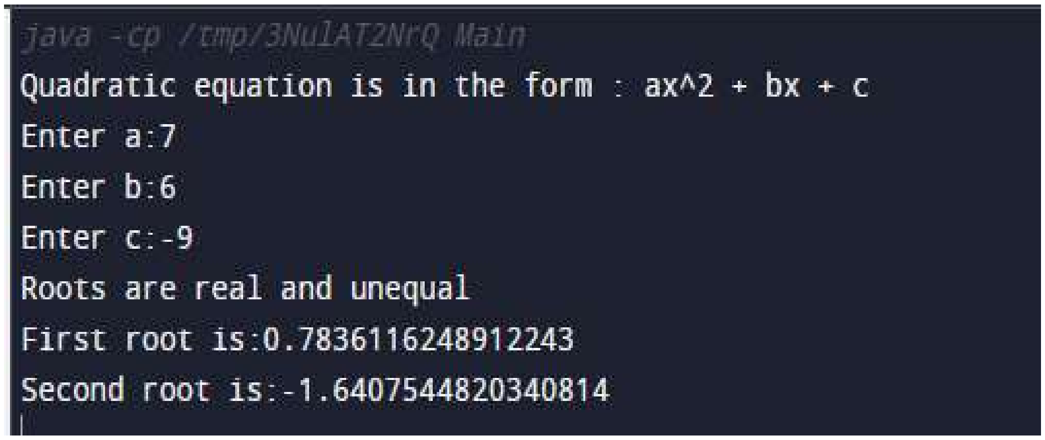
```
java -cp /tmp/uBudYMEzMe GDemo
Name  Phone number  Total
Rama 8060302010 410.0

Name  Phone number  Total
Shama 7689632510 800.0

Name  Phone number  Total
Bhama 9632587412 205.0
```

3. Write a Java program to calculate roots of a quadratic equation. Use appropriate methods to take input, and calculate the roots.

Code:

```
import java.util.Scanner;
class Quad {
int a, b, c;
```

```
double root1, root2, d;
Scanner s = new Scanner(System.in);
void input()
{
System.out.println("Quadratic equation is in the form : ax^2 + bx + c");
 System.out.print("Enter a:");
 a = s.nextInt();
 System.out.print("Enter b:");
 b = s.nextInt();
 System.out.print("Enter c:");
 c = s.nextInt();
}
void discriminant() {
d= (b*b)-(4*a*c);
}
void calculateRoots() {
if(d>0)
{
System.out.println("Roots are real and unequal");
root1 = ( - b + Math.sqrt(d))/(2*a);
root2 = (-b - Math.sqrt(d))/(2*a);
System.out.println("First root is:"+root1);
System.out.println("Second root is:"+root2);
}
else if(d == 0)
 {
 System.out.println("Roots are real and equal");
 root1 = (-b+Math.sqrt(d))/(2*a);
 System.out.println("Root:"+root1);
 }
else
 {
 System.out.println("No real solutions. Roots are imaginary");
 double real = -b / (2 * a);
 double imaginary = Math.sqrt(-d) / (2 * a);
 System.out.println("The equation has two complex roots: " + real + " + " +
imaginary + "i and "
+ real + " - " + imaginary + "i");
 }
}
}
class Main {
public static void main(String[] args) {
Quad q= new Quad();
q.input();
q.discriminant();
q.calculateRoots();
}
}
```

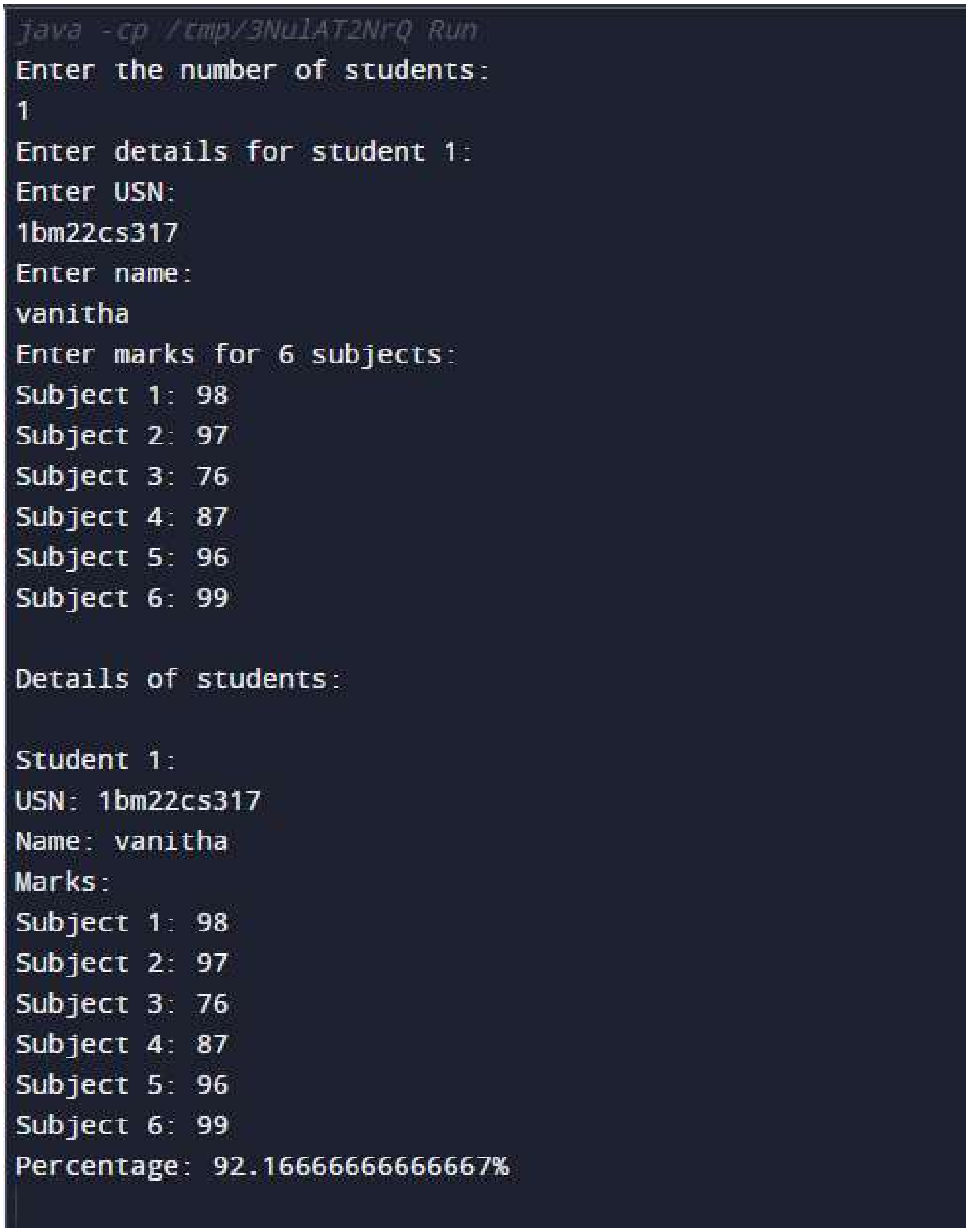


```
java -cp /tmp/3NulAT2NrQ Main
Quadratic equation is in the form : ax^2 + bx + c
Enter a:7
Enter b:6
Enter c:-9
Roots are real and unequal
First root is:0.7836116248912243
Second root is:-1.6407544820340814
```

4. Write a Java program to create a class Student with members USN, name, marks(6 subjects). Include methods to accept student details and marks, Also include a method to calculate the percentage and display appropriate details. (Array of student object to be created)

Code:

```
import java.util.Scanner;

class Student {
    String USN;
    String name;
    int[] marks = new int[6];

    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter USN:");
        USN = scanner.nextLine();
        System.out.println("Enter name:");
        name = scanner.nextLine();
        System.out.println("Enter marks for 6 subjects:");
        for (int i = 0; i < 6; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }

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

    public void displayDetails() {
        System.out.println("USN: " + USN);
        System.out.println("Name: " + name);
        System.out.println("Marks:");
        for (int i = 0; i < 6; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }
        System.out.println("Percentage: " + calculatePercentage() + "%");
    }
}

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

        System.out.println("Enter the number of students:");
        int numStudents = scanner.nextInt();

        Student[] students = new Student[numStudents];

        for (int i = 0; i < numStudents; i++) {
            System.out.println("Enter details for student " + (i + 1) + ":");
            students[i] = new Student();
            students[i].acceptDetails();
        }

        System.out.println("\nDetails of students:");
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            students[i].displayDetails();
        }
    }
}
```

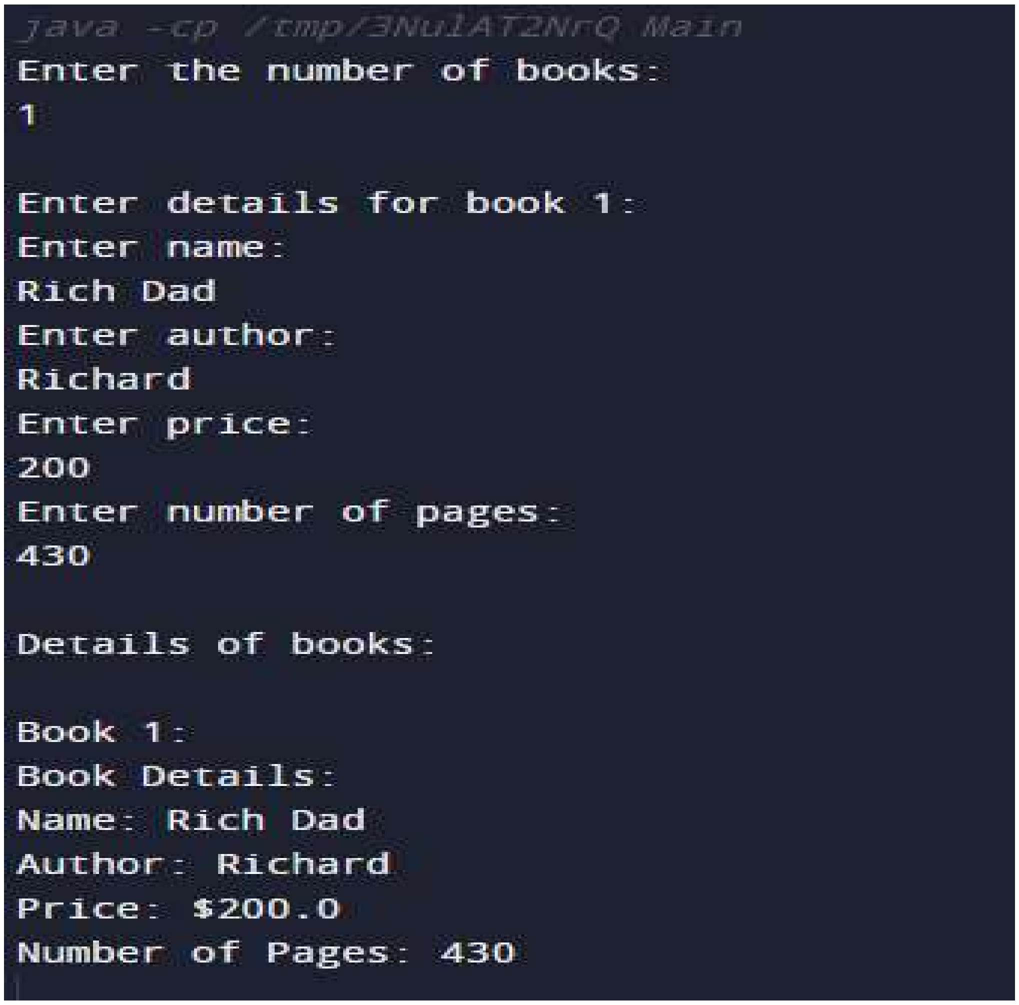


```
java -cp /tmp/3NulAT2NrQ Run
Enter the number of students:
1
Enter details for student 1:
Enter USN:
1bm22cs317
Enter name:
vanitha
Enter marks for 6 subjects:
Subject 1: 98
Subject 2: 97
Subject 3: 76
Subject 4: 87
Subject 5: 96
Subject 6: 99

Details of students:

Student 1:
USN: 1bm22cs317
Name: vanitha
Marks:
Subject 1: 98
Subject 2: 97
Subject 3: 76
Subject 4: 87
Subject 5: 96
Subject 6: 99
Percentage: 92.16666666666667%
```

5.Create a class Book that contains four members: name, author, price, and num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

Code:

```
import java.util.Scanner;

class Book {
    private String name;
    private String author;
    private double price;
    private int num_pages;

    public Book(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }
    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public int getNumPages() {
        return num_pages;
    }

    public void setNumPages(int num_pages) {
        this.num_pages = num_pages;
    }

    @Override
    public String toString() {
        return "Book Details: \n" +
                "Name: " + name + "\n" +
                "Author: " + author + "\n" +
                "Price: $" + price + "\n" +
                "Number of Pages: " + num_pages;
    }
}

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

        System.out.println("Enter the number of books:");
        int numBooks = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        Book[] books = new Book[numBooks];
        for (int i = 0; i < numBooks; i++) {

            System.out.println("\nEnter details for book " + (i + 1) + ":");
            System.out.println("Enter name:");

            String name = scanner.nextLine();
            System.out.println("Enter author:");

            String author = scanner.nextLine();
            System.out.println("Enter price:");

            double price = scanner.nextDouble();
            System.out.println("Enter number of pages:");

            int numPages = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character

            books[i] = new Book(name, author, price, numPages);
        }

        System.out.println("\nDetails of books:");
        for (int i = 0; i < numBooks; i++) {

            System.out.println("\nBook " + (i + 1) + ":");
            System.out.println(books[i]);
        }
    }
}
```

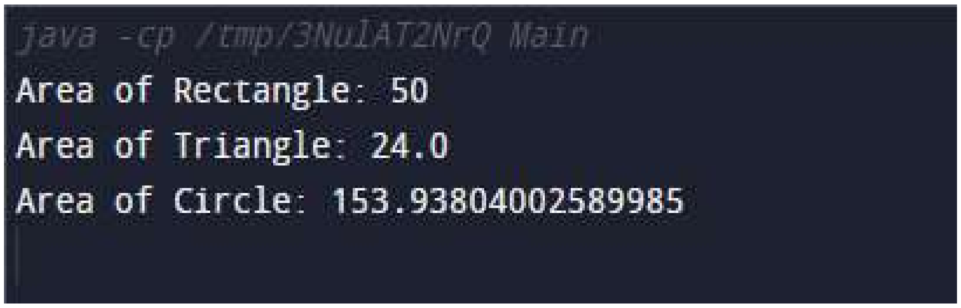


```
java -cp /tmp/3NulAT2NrQ Main
Enter the number of books:
1

Enter details for book 1:
Enter name:
Rich Dad
Enter author:
Richard
Enter price:
200
Enter number of pages:
430

Details of books:

Book 1:
Book Details:
Name: Rich Dad
Author: Richard
Price: $200.0
Number of Pages: 430
```

6. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain the method printArea that prints the area of the given shape.

Code:

```
abstract class Shape {
    protected int dimension1;
    protected int dimension2;

    public Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }

    // Abstract method to calculate and print area
    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        super(length, width);
    }

    @Override
    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }

    @Override
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0); // The second dimension is not needed for a circle
    }

    @Override
    public void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(5, 10);
        Triangle triangle = new Triangle(6, 8);
        Circle circle = new Circle(7);

        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}
```



```
java -cp /tmp/3NulAT2NrQ Main
Area of Rectangle: 50
Area of Triangle: 24.0
Area of Circle: 153.93804002589985
```

7. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

Accept deposit from customer and update the balance.

Display the balance.

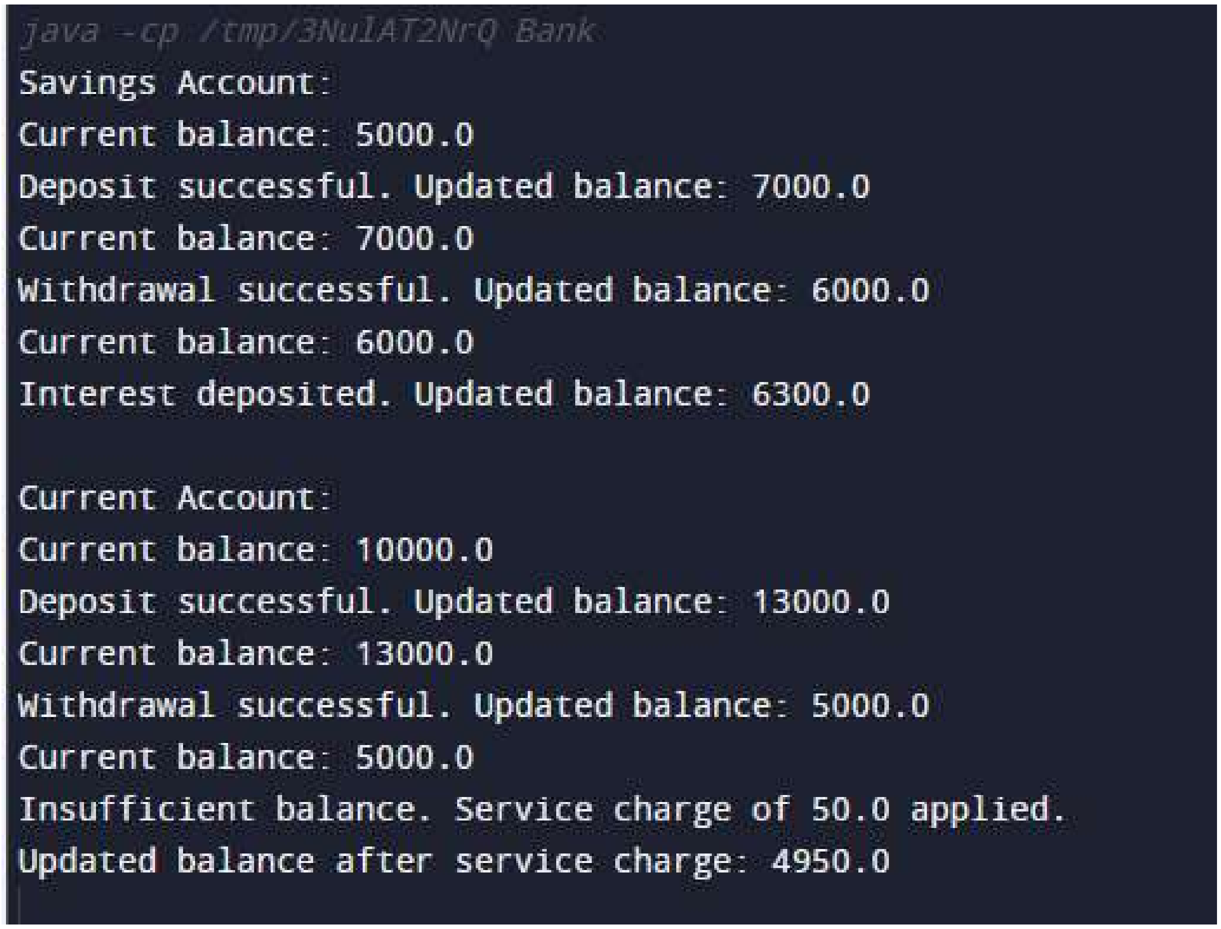
Compute and deposit interest

Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

Code:

```
import java.util.Scanner;

class Account {
    protected String customerName;
    protected long accountNumber;
    protected String accountType;
    protected double balance;

    public Account(String customerName, long accountNumber, String
accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. Updated balance: " + balance);
    }


    public void displayBalance() {
        System.out.println("Current balance: " + balance);
    }
}


class SavingsAccount extends Account {
    private double interestRate;

    public SavingsAccount(String customerName, long accountNumber, String
accountType, double balance, double interestRate) {
        super(customerName, accountNumber, accountType, balance);
        this.interestRate = interestRate;
    }


    public void depositInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest deposited. Updated balance: " + balance);
    }


    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " +
balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}


class CurrentAccount extends Account {
    private double minimumBalance;
    private double serviceCharge;

    public CurrentAccount(String customerName, long accountNumber, String
accountType, double balance, double minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, accountType, balance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }


    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " +
balance);
        } else {
            System.out.println("Insufficient balance. Service charge of " +
serviceCharge + " applied.");
            balance -= serviceCharge;
            System.out.println("Updated balance after service charge: " + balance);
        }
    }
}

class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        SavingsAccount savingsAccount = new SavingsAccount("John Doe",
123456789, "Savings", 5000, 5);


        CurrentAccount currentAccount = new CurrentAccount("Jane Smith",
987654321, "Current", 10000, 2000, 50);


        System.out.println("Savings Account:");
        savingsAccount.displayBalance();
        savingsAccount.deposit(2000);
        savingsAccount.displayBalance();
        savingsAccount.withdraw(1000);
        savingsAccount.displayBalance();
        savingsAccount.depositInterest();


        System.out.println("\nCurrent Account:");
        currentAccount.displayBalance();
        currentAccount.deposit(3000);
        currentAccount.displayBalance();
        currentAccount.withdraw(8000);
        currentAccount.displayBalance();
        currentAccount.withdraw(5000);
    }
}
```



```
java -cp /tmp/3NulAT2NrQ Bank
Savings Account:
Current balance: 5000.0
Deposit successful. Updated balance: 7000.0
Current balance: 7000.0
Withdrawal successful. Updated balance: 6000.0
Current balance: 6000.0
Interest deposited. Updated balance: 6300.0

Current Account:
Current balance: 10000.0
Deposit successful. Updated balance: 13000.0
Current balance: 13000.0
Withdrawal successful. Updated balance: 5000.0
Current balance: 5000.0
Insufficient balance. Service charge of 50.0 applied.
Updated balance after service charge: 4950.0
```

8.Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class internals derived from student has an array that stores the internal marks scored in five 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 five 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 five courses.

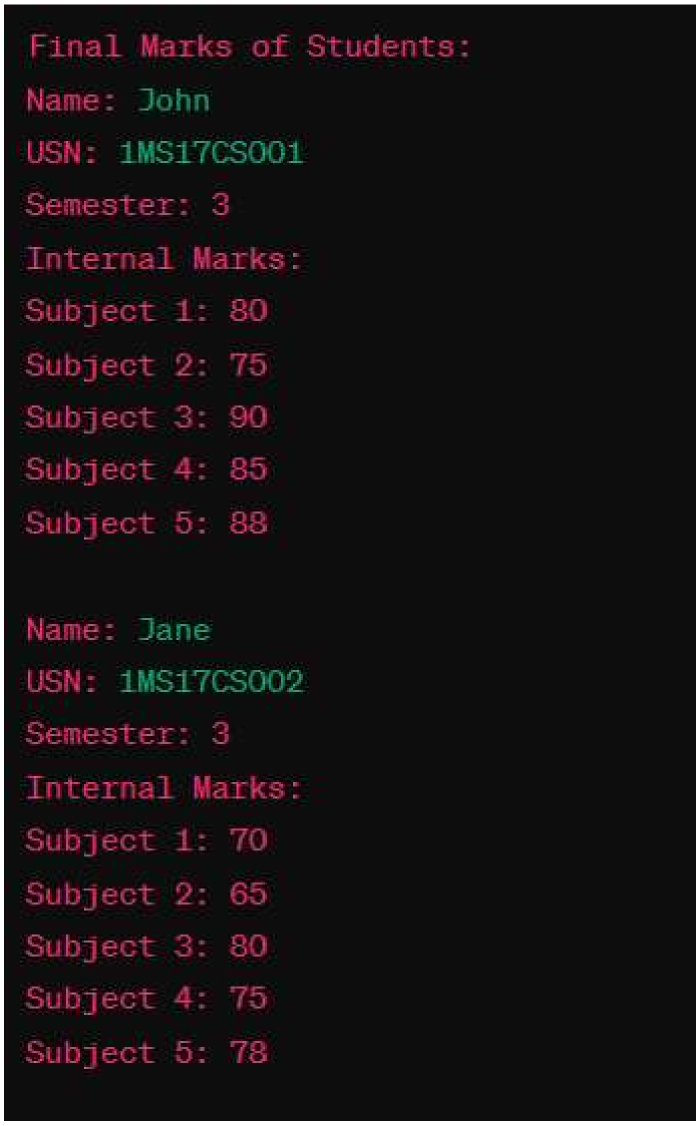
1. Create a folder CIE and save the programs Student.java and Internals.java within it.
2. Create a folder SEE and save the program External.java within it.
3. Save the Main program outside these two folders.
4. Compile Main.java and Execute the Main.class

Code:

```
package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;

    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
package CIE;

public class Internals extends Student {
    public int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}
package SEE;
import CIE.Student;

public class External extends Student {
    public int[] externalMarks;

    public External(String usn, String name, int sem, int[] externalMarks) {
        super(usn, name, sem);
        this.externalMarks = externalMarks;
    }
}
import CIE.Internals;
import SEE.External;
```

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

        int[] internalMarks1 = {80, 75, 90, 85, 88};
        Internals student1 = new Internals("1MS17CS001", "John", 3,
internalMarks1);

        int[] internalMarks2 = {70, 65, 80, 75, 78};
        Internals student2 = new Internals("1MS17CS002", "Jane", 3,
internalMarks2);


        System.out.println("Final Marks of Students:");
        displayFinalMarks(student1);
        displayFinalMarks(student2);

    }

    public static void displayFinalMarks(Student student) {
        System.out.println("Name: " + student.name);
        System.out.println("USN: " + student.usn);
        System.out.println("Semester: " + student.sem);

             if (student instanceof Internals) {
            Internals internalsStudent = (Internals) student;
            System.out.println("Internal Marks:");
            for (int i = 0; i < internalsStudent.internalMarks.length; i++) {
                System.out.println("Subject " + (i + 1) + ": " +
internalsStudent.internalMarks[i]);
            }
        } else if (student instanceof External) {
            External externalStudent = (External) student;
            System.out.println("External Marks:");
            for (int i = 0; i < externalStudent.externalMarks.length; i++) {
                System.out.println("Subject " + (i + 1) + ": " +
externalStudent.externalMarks[i]);
```

```
            }
        }

        System.out.println();
    }
}
```



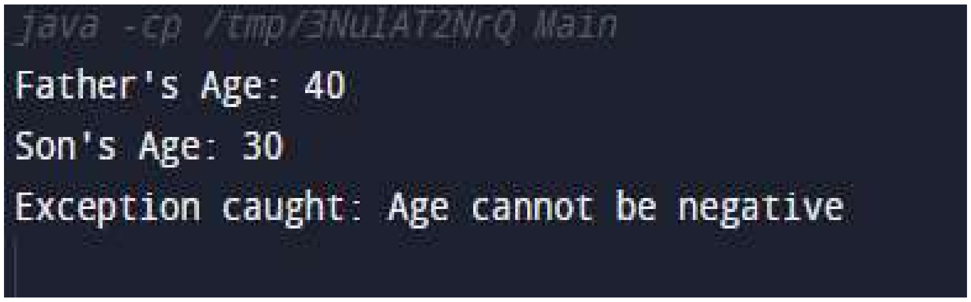
```
Final Marks of Students:
Name: John
USN: 1MS17CS001
Semester: 3
Internal Marks:
Subject 1: 80
Subject 2: 75
Subject 3: 90
Subject 4: 85
Subject 5: 88

Name: Jane
USN: 1MS17CS002
Semester: 3
Internal Marks:
Subject 1: 70
Subject 2: 65
Subject 3: 80
Subject 4: 75
Subject 5: 78
```

9.Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

Code:

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

class Father {
    private int age;

    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to
father's age");
```

```
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
    }
}

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

            Son son = new Son(40, 30);
            System.out.println("Father's Age: " + son.getAge());
            System.out.println("Son's Age: " + son.getSonAge());
        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        }

        try {

            Son son2 = new Son(-10, 20);
            System.out.println("Father's Age: " + son2.getAge());
            System.out.println("Son's Age: " + son2.getSonAge());
        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

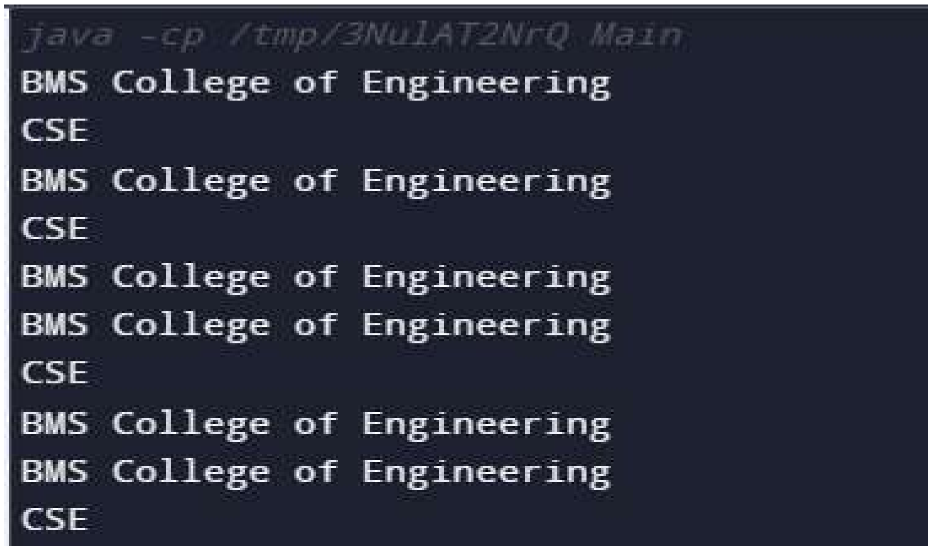


```
java -cp /tmp/3NulAT2NrQ Main
Father's Age: 40
Son's Age: 30
Exception caught: Age cannot be negative
```

10.Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

Code:

```
class BMSCEThread extends Thread {
    public void run() {
        while (true) {
            System.out.println("BMS College of Engineering");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
class CSEThread extends Thread {
    public void run() {
        while (true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
public class Main {
    public static void main(String[] args) {
        BMSCEThread bmsceThread = new BMSCEThread();
        CSEThread cseThread = new CSEThread();
        bmsceThread.start();
        cseThread.start();
    }
}
```

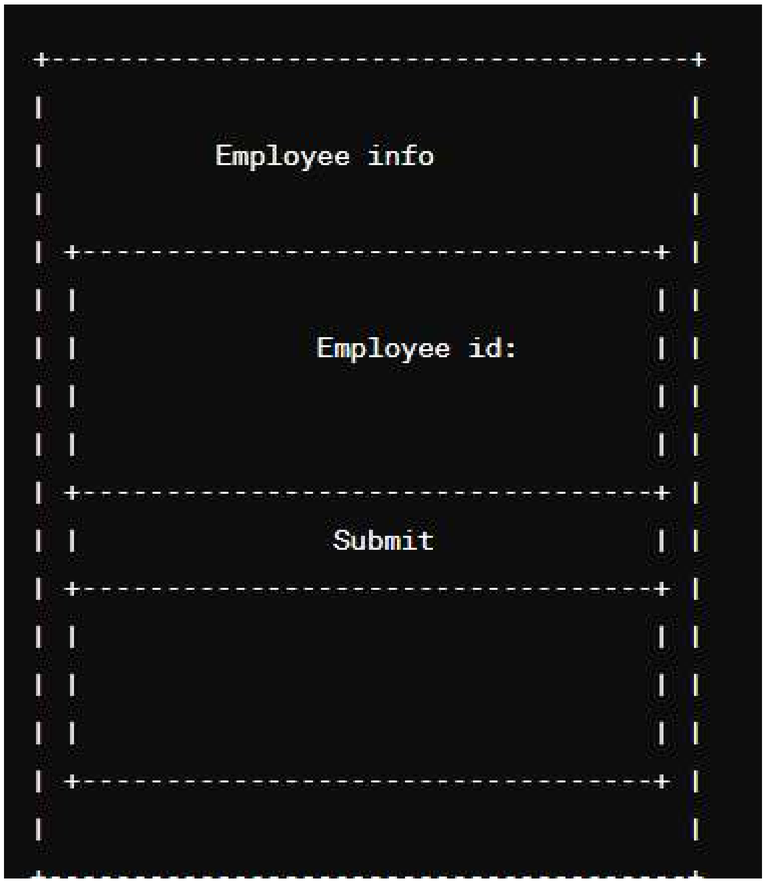


```
java -cp /tmp/3NulAT2NrQ Main
BMS College of Engineering
CSE
BMS College of Engineering
CSE
BMS College of Engineering
BMS College of Engineering
CSE
BMS College of Engineering
BMS College of Engineering
CSE
```

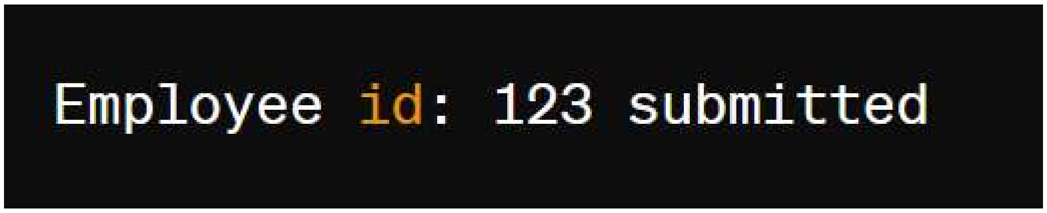
11.Creating label, button and TextField in a Frame using AWT.

code:

```
import java.awt.*;
import java.awt.event.*;



public class AWTExample extends WindowAdapter {

    Frame f;

    AWTExample() {
        f = new Frame();
        f.addWindowListener(this);

        Label l = new Label("Employee id:");
        Button b = new Button("Submit");
        TextField t = new TextField();

        l.setBounds(20, 80, 80, 30);
        t.setBounds(20, 100, 80, 30);
        b.setBounds(100, 100, 80, 30);

        f.add(b);
        f.add(l);
        f.add(t);

        f.setSize(400, 300);
        f.setTitle("Employee info");
        f.setLayout(null);
        f.setVisible(true);
    }

    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }

    public static void main(String[] args) {
        AWTExample obj = new AWTExample();
    }
}
```



```
+-----------------------------------------+
|                                         |
|           Employee info                 |
|                                         |
| +-----------------------------------+   |
| |                                   | |
| |             Employee id:          | |
| |                                   | |
| |                                   | |
| +-----------------------------------+   |
| |              Submit               | |
| +-----------------------------------+   |
| |                                   | |
| |                                   | |
| |                                   | |
| +-----------------------------------+   |
|                                         |
+-----------------------------------------+
```



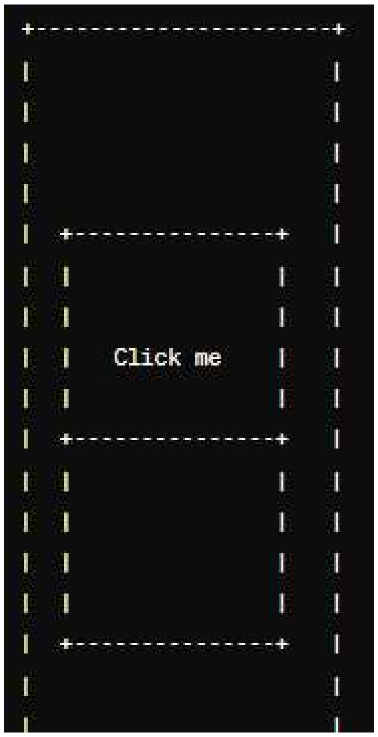
```
Employee id: 123 submitted
```

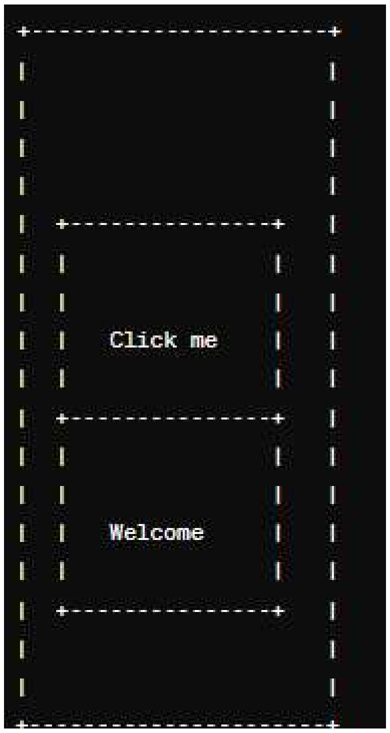
12. Create a button and add a action listener for Mouse click.

Code:

```
import java.awt.*;
import java.awt.event.*;

class AWTExample extends WindowAdapter implements ActionListener {

    Frame f;
    TextField t;
```

```
AWTExample() {
    f = new Frame();
    f.addWindowListener(this);

    Label l = new Label("Employee id:");
    Button b = new Button("Submit");
    t = new TextField();

    l.setBounds(20, 80, 80, 30);
    t.setBounds(20, 100, 80, 30);
    b.setBounds(100, 100, 80, 30);

    b.addActionListener(this);

    f.add(b);
    f.add(l);
    f.add(t);

    f.setSize(400, 300);
    f.setTitle("Employee info");
    f.setLayout(null);
    f.setVisible(true);
}

public void windowClosing(WindowEvent e) {
    System.exit(0);
}

public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("Submit")) {
        System.out.println("Employee id: " + t.getText() + " submitted");
    }
}

public static void main(String[] args) {
    AWTExample obj = new AWTExample();
    }
}
```

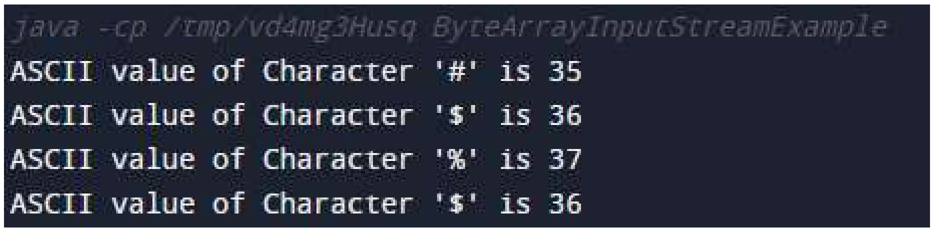




13. I/O Package

Code 1:

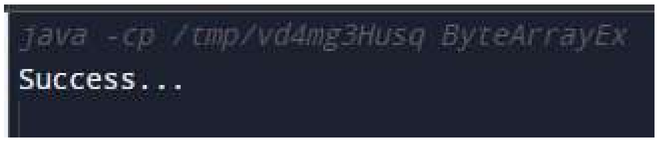
```
import java.io.*;

public class ByteArrayInputStreamExample {

    public static void main(String[] args) throws IOException {
        byte[] buf = {35, 36, 37, 36};

        ByteArrayInputStream byt = new ByteArrayInputStream(buf);

        int val;
        while ((val = byt.read()) != -1) {
            char ch = (char) val;
            System.out.println("ASCII value of Character '" + ch + "' is " + val);
        }

        FileInputStream fin = new FileInputStream("example.txt");
        System.out.print("Content from file: ");
        int data;
        while ((data = fin.read()) != -1) {
            System.out.print((char) data);
        }
        fin.close();
    }
}
```



```
java -cp /tmp/vd4mg3Husq ByteArrayInputStreamExample
ASCII value of Character '#' is 35
ASCII value of Character '$' is 36
ASCII value of Character '%' is 37
ASCII value of Character '$' is 36
```

Code 2:

```
import java.io.*;

public class ByteArrayEx {

    public static void main(String[] args) throws Exception {
        FileOutputStream fout1 = new FileOutputStream("Example.txt");
        FileOutputStream fout2 = new FileOutputStream("Example2.txt");

        ByteArrayOutputStream bout = new ByteArrayOutputStream();
        bout.write(65);

        bout.writeTo(fout1);
        bout.writeTo(fout2);

        bout.flush();
        bout.close();
        fout1.close();
        fout2.close();

        System.out.println("Success...");
    }
}
```



```
java -cp /tmp/vd4mg3Husq ByteArrayEx
Success...
```