Assignment-1

Q.1 Program to print prime numbers from 1 to 100.

→

```
public class Prime_Number {
    public static void main (String[] args)
    {
        System.out.println("Prime numbers from 1 to 100 are :");
        for (int i = 1; i <= 100; i++)
        {
            int count=0;
            for(int num =i; num>=1; num--)
            {
                if(i%num==0)
                {
                    count = count + 1;
                }
            }
            if (count ==2)
                System.out.print(i+" ");
        }
    }
}
```

Q.2 Program to find and print the factorial of a given number

→

```
import java.util.Scanner;

public class Factorial
{
  public static void main(String[] args)
 {
  Scanner s= new Scanner(System.in);
  System.out.println("Enter the Number : ");
  long num = s.nextLong();
  for (long i=(num-1);i>=2;i--)
    num=num*i;
  System.out.println(num);
 }
}
```

Q.3 Program to print the following pattern.

1

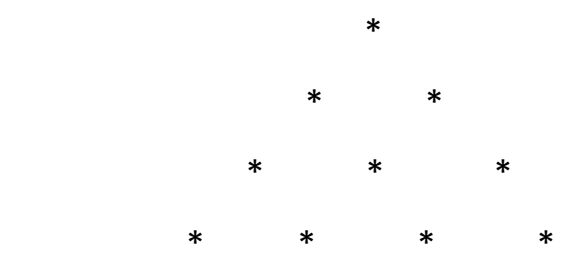
1 2

1 2 3

➔

```
public class Pattern {
    public static void main(String[] args) {
        for (int i=1;i<=3;i++)
        {
            for (int j=1;j<=i;j++)
            {
                System.out.print(j+" ");
            }
            System.out.println();
        }
    }
}
```

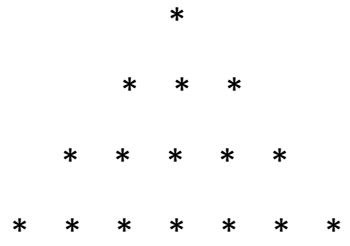
Q.4 Program to print the following pattern



→

```
public class Pattern2 {
    public static void main(String[] args) {
        for (int i=1;i<=4;i++)
        {
            for (int j=4-i;j>=1;j--)
            {
                System.out.print("    ");
            }
            for (int k=1;k<=i;k++)
            {
                System.out.print("  *  "+"  ");
            }
            System.out.println();
        }
    }
}
```

Q.5 Program to print following pattern:



→

```
public class Pattern3 {
    public static void main(String[] args) {
        for (int i=1;i<=4;i++)
        {
            for (int j=4-i;j>=1;j--)
            {
                System.out.print("   ");
            }
            for (int k=1;k<=(i*2)-1;k++)
            {
                System.out.print(" * ");
            }
            System.out.println();
        }
    }
}
```

Assignment-2

Q.1 Write a program to enter a password and check length not < 8 and > 15.

→

```
import java.util.Scanner;

public class Password {
    public static void main(String[] args) {
        Scanner s= new Scanner(System.in);
        System.out.println("Enter Password:");
        String password = s.nextLine();
        if (password.length()>8 && password.length()<15)
            System.out.println("Strong Password!");
        else
            System.out.println("Poor Password!");
    }
}
```

Q.2 Program to check password and repassword are same or not :

→

```
import java.util.Scanner;
public class password2 {
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter your password:");
        String password = s.nextLine();
        System.out.println("Re-enter Password:");
        String matches = s.nextLine();
        if (password.equals(matches))
            System.out.println("Password is correct");
        else
            System.out.println("Password didn't matched");
    }
}
```

Assignment-3

Q.1 Program to check occurrence of character in a sentence using lastIndexOf() method.

→

```
import java.util.Scanner;

public class Assignment3 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter a Sentence : ");
        String sentence = s.nextLine();
        System.out.println("Enter a character to Search:");
        String character = s.nextLine();
        int count = 0;
        int pos = sentence.length();
        while ((pos=sentence.lastIndexOf(character,pos))!=-1)
        {
            count++;
            pos--;
        }
        System.out.println(character + " occurs " + count + " times. ");
    }
}
```

Q.2 Program to check whether the string is palindrome or not.

→

```
import java.util.Scanner;
public class palandrome {
public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    System.out.println("Enter a String : ");
    String string1 = s.nextLine();
    StringBuffer string2 = new StringBuffer(string1);
    string2.reverse();
    if(string1.equals(string2.toString()))
    {
        System.out.println(string1 + " is palandrome");
    }
    else
        System.out.println(string1 + " is not a palandrome");
}
}
```

Assignment 4

Q.1 Create a class Student with following members:

1) Instance variable : String name , result; double eng, math, science, tmarks.per;
2) Constructor : Strudent(String, double, double, double)
3) Methods : void setData(String, double, double, double), void calResult(), void showResult()

❼

```
import java.util.Scanner;
class Student{     String
Name, Result;
   double English, Math, Science, Total_Marks, Percentages;

   Student(String name, double english, double math, double science){
       Name=name;
       English=english;
       Math=math;
       Science=science;
   }
   void setData(String name, double english, double math, double science ){
      Name=name;
      English=english;
      Math=math;
      Science=science;
   }
   void CalResult(){
```

```
    Total_Marks = English+Math+Science;
Percentages = (Total_Marks/300)*100;
if(Percentages>=80)         Result = "1st
Class";
    if(Percentages>=60 && Percentages<=80)
Result = "2nd Class";
    if(Percentages>=40 && Percentages<=60)
Result = "3rd Class";      if(40>=Percentages)
        Result = "Fail";
   }
   void ShowResult(){
       System.out.println("Total Marks : "+Total_Marks);
       System.out.println("Percantages : "+Percentages);
       System.out.println("Result : "+Result);
   }
}
public class Assignment_4 {     public
static void main(String[] args) {
       Scanner s = new Scanner(System.in);
       System.out.println("Enter Student's Name :");
       String Name = s.next();
       System.out.println("Enter Marks of the Following Subjects :
\n1.English\n2.Math\n3.Science");
double English = s.nextDouble();
double Math = s.nextDouble();
double Science = s.nextDouble();
```

```
Student s1 = new Student(Name , English,
Math, Science );        s1.CalResult();
s1.ShowResult();
   }
}
```

Assignment - 5

Q.1 Difference between Method Overloading and Method Overloading

→

Method Overloading	Method Overriding
• Method Overloading is used to increase the readiability of the progrcem.	• Method Overriding is used to provide the specific implementation of the method that is already provided by it's super class.
• It is performed within class.	• It occurs in two classes that have a (inheritance) relationship.
• Parcameter must be diff^{n}	• Parcameter must be same.
• It is example of compile time polymorphism	• It is the example of run time polymorphism
• Return tupe can be same or different in method overloading but you must have to change the parcameter	• Return tupe must be same or covarient in method overriding

Scanned with CamScanner

Q.2 • Create a class Employee :

Instance variable : empno, ename, basic, netsalary

Constructors : Use neccessary constructor to initialize

Method : calcsalary() to calculate netsalary.

• Create a class manager inheriting employee class

Instance variable : hra, da, ta assume da = 50% of basic

hra, ta ← pass from cmd prompt

Method : override method calcsalary() to calculate netsalary for manager class.

→

```
import java.util.Scanner;
class Employee {
        string ename;
        Float empno, basic, netsalary;
        Employee () { }
        Employee ( string name, Float no) {
            ename = name;
            empno = no;
        }
      void calcsalary () {
            Scanner a = new Scanner (System.in);
            System.out.println ("Enter basic Salary");
            basic = a.nextFloat();
            System.out.println ("Net-salary of
                Employee is " + basic);
}
```

Scanned with CamScanner

```
class Manager extends employee {
    float hra, da, ta;
    Manager() { }
    Manager (sbing name, float no) {
        ename = name;
        empno = no;
    }
    void calcsalary () {
        Scanner b = new Scanner(System.in);
        System.out.println(" Enter basic salary, hra
                            and ta");
        basic.= b.nextFloat ();
        hra = b.nextfloat();
        ta = b.nextFloat();
        da = basic / 2;
        netsalary = hra + ta + da + basic;
        System.out.println(" Net salary of
                            manager is " + netsalary); }
}
public class main {
    public static void main (string[].args) {
        Employee e = new employee("Bhavin",
                                  1807039);
        e.netsalary ();
        Manager m = new manager(" Blackhat",
                                1807001);
        m.calcsalary ();
    }
}
```

Scanned with CamScanner

Q.3 On applying Method Overriding on class figure's Method double area(), Class: Rectangle, Triangle by extending figure class.

→

```
class figure {
    double dia1, dia2;
    figure () { }
    figure ( double d1 , double d2) {
        dia1 = d1;
        dia2 = d2;
    }
    double area () {
        return 0;
    }
}
class Rectangle extends figure {
    Rectangle () { }
    Rectangle ( double d1, double d2) {
        dim1 = d1;
        dia2 = d2;
    }
    double area () {
        return dia1* dia2;
    }
}
class Triangle extends figure {
    Triangle () { }
    Triangle ( double d1, double d2) {
        dia1= d1;
        dia2= d2;
    }
```

Scanned with CamScanner

```
    double area(){
        return 0.5*dia1*dia2;
    }
}
public class main(){
    public static void main(String args[])
    {
        Rectangle r = new Rectangle(30, 30);
        Triangle t = new Triangle(45, 45);
        double rectangleArea = r.area();
        double triangleArea = t.area();
        System.out.println("Area of Rectangle = "
                + rectangleArea);
        System.out.println("Area of Triangle = "
                + triangleArea);
    }
}
```

Scanned with CamScanner

Assignment No. 6

Q.1 Modify class figure as an abstract class by adding method.

→

```
import java.util.Scanner;
abstract class figure{
        double dia1, dia2;
        figure(){}
        abstract double area();
}
class Triangle extends figure{
        Triangle(){}
        Triangle(double d1, double d2){
                dia1 = d1;
                dia2 = d2;
        }
        double area(){
                return 0.5* dia1 * dia2;
        }
}
public class Assignment_6{
        public static void main(String args[]){
                figure rect, tri;

class Rectangle extends figure{
        Rectangle(){}
        Rectangle(double d1, double d2){
                dia1 = d1;
                dia2 = d2;
        }
```

Scanned with CamScanner

```
    double area(){
        return dia1 * dia2;
    }
}
public class Assignment6_1{
    public static void main(String args[]){
        Figure triangle, rectangle;
        triangle = new Triangle(60, 60);
        rectangle = new Rectangle(40, 40);
        System.out.println("Area of Triangle = "
                + triangle.area());
        System.out.println("Area of Rectangle = "
                + rectangle.area());
    }
}
```

Q.2 Create interface vehicle and implement splendor from vehicle. Create another interface VehicleNew and implement Pulsor from VehicleNew.

→

```
interface Vehicle {
    int price();
    int avg();
    int cc();
}
```

Scanned with CamScanner

```
class Splender implements Vehicle {
        String model;
        Splender(String m) { model = c; }
        public int price() { return 28,000; }
        public int avg() { return 55; }
        public int cc() { return 55; }
        void showModel(){
            system.out.println("Model is " + model);
        }
}
interface newVehicle(){
        int price();
        int avg();
        int cc();
}
class Pulsar implements newVehicle{
        String model;
        Pulsar(String m){
            model = m;
        }
        public int price(){ return 32,000; }
        public int avg(){ return 55; }
        public int cc(){ return 55; }
        void showModel(){
            System.out.println("Model is " + model);
        }
}
```

Scanned with CamScanner

```
class Assignment6_2 {
    static void showSplendor(Splendor s) {
        System.out.println("Price : " + s.price());
        System.out.println("Avgrage : " + s.avg());
        System.out.println("CC : " + s.cc());
    }
    static void showPulsar(Pulsar p){
        System.out.println("Price : " + p.price());
        System.out.println("Average : " + p.avg());
        System.out.println("CC : " + p.cc());
    }
    public static void main(String args[]){
        Splendor s = new Splendor("Splendor_
                                   Plus");
        Pulsar p = new Pulsar("Bajaj");
        showSplendor(s);
        showPulsar(p);
        s.showModel();
        System.out.println("----------");
        showPulsar(p);
        p.showModel();
    }
}
```

Scanned with CamScanner

Assignment 8.

Q.1 Create a class ChildThread extending class Thread or implementing Runnable interface. Use necessary constructors to initialize. From it's run if thread name is odd then print odd numbers or if thread name is even then print even numbers. Create a class RunnableDemo/ThreadDemo with public stetic void main. Inside main create two threads. Odd and Even should be passed as thread names to the ChildThreads.

→

```
Import java.lang.Thread;

class childThread implements Runnable {
        Thread t;
        childThread (String s){
                t = new Thread (this, s);
                run();
                System.out.println("Constructor exiting...");
        }
        public void run() {
                String child = t.getName();
                System.out.println("Thread Name : "+ child);
                int i = 1;
                if (child.equals("Odd"))
                i = 1;
                else if (child.equals("Even"))
                i = 0;
                else return;
                try {
```

Scanned with CamScanner

```
            for (; i<=10; i=i+2){
                System.out.println("Child Thread :"
                                    +child+"->"+i);
                t.sleep(500);
            }
        }
        catch(Exception e) { }
        System.out.println(child+" Thread is exiting...");
    }
}

public class Assignment8_1{
    public static void main(String args[]){
        String name = Thread.currentThread().
                                    getName();
        childThread odd = new childThread("Odd");
        childThread even = new childThread("Even");
    }
}
```

Q.2. • Create a class Bank with instance variable: balance, Use neccessary constructor to initialize balance. Define a synchronized method: withdraw(int amt) {.....}

- The withdraw method should check if balance is sufficient. IF sufficient then display message "Your Transaction is under process" and call sleep method for 2 seconds.
- Next update balance (i.e. balance = balance - withdraw-amt)
- Next display message "Your Transaction is completed: Avail balance is:"
- Call sleep for 1 second and then display "Thank You Message!
- Next create a BankThread class. from it's run() call Bank withdraw().

Scanned with CamScanner

• Next create BankDemo with public static void main().
• Inside main create Bank Object. Next Try to perform 4 types of transaction: Cash, Cheque, DD and ATM
• (cash, cheque, DD ATM should be the names passed as thread names to the BankThread class.

→

```
import java.lang.Thread;
class Bank {
    double balance;
    Bank(){}
    Bank(double b)
    { balance = b; }

    synchronized public void withdraw(double wa){
        String n = Thread.currentThread().getName();
        if(balance >= wa){
            System.out.println("Your "+n+" transaction is
                                under process....");
            try {
                Thread.sleep(2000);
                balance = balance - wa;
                System.out.println("Transaction
                                Completed");
                System.out.println("Withdraw amount="
                                +wa+"Rs");
                System.out.println("Collect your cash \n
                    Balance : "+balance);
                Thread.sleep(1000);
                System.out.println("Thank you!");
            }
            catch(Exception e){}
        }
```

Scanned with CamScanner

```
            else {
                System.out.println("Transaction failed:" +n);
                System.out.println("Insufficient balance:"
                                        +balance);
            }
        }
}

class BankThread implements Runnable {
    Bank ref;
    double wa;
    BankThread (Bank ref, double wa, String name) {
        this.ref = ref;
        this.wa = wa;
        new Thread (this, name).start();
    }
    public void run() {
        ref.withdraw(wa);
    }
}
public class Assignment8_2 {
    public static void main(String args[]) {
        Bank b1 = new Bank(100000);
        BankThread cash = new BankThread(b1,
                                10000, "Cash");
        BankThread cheque = new BankThread(b1,
                                20000, "Cheque");
        BankThread atm = new BankThread(b1,
                                20000, "ATM");
        BankThread dd = new BankThread(b1, 40000,
                                "DD");
    }
}
```

Scanned with CamScanner

Assignment 9

Q.1 Create a class ByteDemo and takes a console input through keyboard using System.in.read().

→
```
class ByteDemo {
public static void main(String args[]) throws
                                java.io.IOException {
        System.out.println("Enter a Character:");
        char n = (char) System.in.read();
        System.out.println("You Entred"+n);
    }
}
```

Q.2 Create a class BufferedReaderDemo and takes a keyboard input through keyboard using Buffered Reader

→
```
import java.io.*;
class BufferedReaderDemo {
    public static void main(String args[]) throws
                                IOException {
        char c;
        InputStreamReader r = new InputStreamReader
                                (System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter Charecter");
        c = (char) br.read();
        System.out.println(c);
    }
}
```

Scanned with CamScanner

Assignment 10

Q.1. Create a class DataInputStreamDemo and takes a keyboard input through keyboard using DataInputStream.

→

```
import java.io.*;
public class DataInputStreamDemo{
        public static void main(String[] args) throws
                                                IOException {
            DataInputStream d1 = new DataInputStream
                                        (System.in);
            System.out.println("Enter your age:");
            int age = Integer.parseInt(d1.readLine());
            System.out.println("Enter your name: ");
            String name = d1.readLine();
            System.out.println("Age :"+age+"Name :"+name);
        }
}
```

Scanned with CamScanner

Assignment 11

Q.1 Create a class SequenceInputStreamDemo, Take four files as input using FileInputStream and pass it in vector of FileInputStream. Using Enumeration read elements from vector and passed it to SequenceInputStream and print it.

```
import java.io.*;
import java.util.*;
public class SequenceInputStreamDemo {
    public static void main(String args[]) throws
                                        Exception {

        FileInputStream file1 = new FileInputStream("f1.txt");
        FileInputStream file2 = new FileInputStream("f2.txt");
        FileInputStream file3 = new FileInputStream("f3.txt");
        FileInputStream file4 = new FileInputStream("f4.txt");
        Vector v = new Vector();
        v.add(file1);
        v.add(file2);
        v.add(file3);
        v.add(file4);
        Enumeration e = v.elements();
        SequenceInputStream s = new SequenceInputStream(e);
        int a;
        while((a = s.read()) != -1)
            System.out.print((char)a);
        s.close();
        file1.close();
        file2.close();
        file3.close();
        file4.close();
    }
}
```

Scanned with CamScanner

Assignment 12

Q.1. Design an applet for an alternate moving banner.

→
```
import java.awt.*;
import java.applet.*;
/* <applet code = "Bhavin's Banner" width = 535
      height = 360> </applet> */

public class Assignment12 extends Applet implement
                                        Runnable {
    String msg = "Bhavin's Moving Banner";
    int x, y, flag;
    boolean stopflag;
    public void init () {
        x = 100;
        y = 100;
        flag = 1;
        Thread t = new Thread (this, "mythread");
        setBackground (color.Black);
        setforeground (
        setforeground (color.red);
        t.start ();
    }
    public void update () {
        x = x + 50 * flag;
        if (x >= 360)
          flag = -1;
        if ( x <= 0)
          flag = 1;
    }
```

Scanned with CamScanner

```
public void run(){
    while (true){
        repaint();
        update();
        try{
            Thread.sleep(1000);
        }
        catch(InterruptedException e){
            System.out.println(e);
        }
    }
}
public void paint(Graphics g){
    g.drawString(msg,x,y);
}
}
```

Scanned with CamScanner

Assignment : 13

Q1. Depending upon how many times mouse button clicked, change background color. Upto 4 times using a switch. Find out which button was clicked. Make use of getModifier. switch(count){case 1: // break; case 2: //.....}

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/* <applet code = "Assignment13" width = "600" height = "600">
    </applet> */

public class Assignment13 extends Applet{
        public void init(){
            Button btn = new Button("Button");
        setLayout(null);
        btn.setBounds(250,200,100,20);
        btn.addMouseListener(new HelperListener(this));
        add(btn);
        setVisible(true);
        }
}
class HelperListener extends MouseAdapter{
        Assignment13 ad;
        public HelperListener(Assignment13 ad){
            this.ad = ad;
        }
```

Scanned with CamScanner

```
public void mouseClicked(MouseEvent e){
    if((e.getModifiers() & InputEvent.BUTTON1_
                                MASK) != 0){
        System.out.println("Left click detected");
    }
    if((e.getModifiers() & InputEvent.BUTTON2_
                                MASK) != 0){
        System.out.println("Middle Click detected");
    }
    if((e.getModifiers() & InputEvent.BUTTON3_MASK)
                                != 0){
        System.out.println("Right Click detected");
    }
    int count = e.getClickCount();
    switch (count) {
        case 1:
            ad.setBackground(Color.Black);
            break;
        case 2:
            ad.setBackground(Color.Yellow);
            break;
        case 3:
            ad.setBackground(Color.red);
            break;
        case 4:
            ad.setbackgoound(Color.blue);
            break;
        case 5:
            ad.setbackground(Color.green);
            break;
        default: break;
    }
}
```

Scanned with CamScanner

Assignment 15

Q.1 Design an applet for calculator use of panel

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/* < applet code = "Assignment15_1" width = 600
    height = 600 > </applet> */

public class Assignment15_1 extends Applet {
    Label l1, l2, l3;
    TextField t1, t2, t3;
    Button addbtn, subbtn, multibtn, divbtn;
    public void init(){
        Panel p1 = new Panel();
        p1.setBounds(30, 40, 200, 80);
        Panel p2 = new Panel();
        p2.setBounds(30, 130, 200, 30);
        p1.setLayout(new GridLayout(3, 2, 8, 10));
        p2.setLayout(new GridLayout(1, 4, 6, 6));
        setLayout(null);
        add(p1);
        add(p2);
        l1 = new Label("First Number: ");
        l2.setBackground
        l2 = new Label("Second Number: ");
        l3 = new Label("Result: ");

        t1 = new TextField(10);
        t2 = new TextField(10);
        t3 = new TextField(10);
```

Scanned with CamScanner

```
t3.setEditable(false);

addbtn = new Button("ADDISION");
subbtn = new Button("SUBSTRACTION");
multibtn = new Button("MULTIPLICATION");
divbtn = new Button("DIVISION");

p1.add(l1); p1.add(t1); p1.add(l2); p1.add(t2);
p1.add(l3); p2.add(t3);
p2.add(addbtn); p2.add(subbtn);
p2.add(multibtn); p2.add(divbtn);

ActionListener listener = new ActionListener()
{
    public void actionPerformed(ActionEvent e){
        String c = e.getActionCommand();
        calculation(c);
    }
};
addbtn.addActionListener(listener);
subbtn.addActionListener(listener);
multibtn.addActionListener(listener);
divbtn.addActionListener(listener);
}
public void calculation(String c){
    t1.select(2,5);
    double a = Double.parseDouble(t1.getText());
    double b = Double.parseDouble(t2.getText());
    double c = 0;
    if(c.equals("ADDISION")) c = a+b;
    t3.setText(c+" ");
```

Scanned with CamScanner

```
        if(c.equals("SUBSTRACTION")) c = a-b;
        t3.setText(c+" ");
        if(c.equals("MULTIPLICATION")) c = a*b;
        t3.setText(c+" ");
        if(c.equals("DIVISION")) c = a/b;
        t3.setText(c+" ");
    }
}
```

Q2. Make use of 3 cards using card layout

→

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code = "Assignment15_2" width = 400
    height = 400> </applet> */

public class Assignment15_2 extends Applet {
        Panel deck, p1, p2, p3;
        CardLayout c = new CardLayout(40,30);
        public void init() {
            deck = new Panel();
            p1 = new Panel();
            p1.setBackground(Color.blue);
            p1.add(new Checkbox("Window 7"));
            p1.add(new Checkbox("Window XP"));
            p2 = new Panel();
            p2.setBackground(Color.red);
            p2.add(new Checkbox("macOS"));
            p2.add(new Checkbox("Linux"));
```

Scanned with CamScanner

```
p3 = new Panel();
p3.setBackground(Color.green);
p3.add(new Checkbox("Unix"));
p3.add(new Checkbox("MS-DOS"));
deck.setLayout(c);
deck.add("Windows", p1);
deck.add("Others", p2);
deck.add("Others and mac", p3);
add(deck);
addMouseListener(new MouseAdapter(){
        public void mouseClicked(MouseEvent e)
        {
            c.next(deck);
        }
});
    }
}
```

Scanned with CamScanner