Part A

1. Program to assign two integer values to X and Y. Using the ‘if’ statement the output of the program should display a message whether X is greater than Y.

```
import java.util.Scanner;
public class GreaterNumber{
      public static void main(String args[]){
            Scanner sc = new Scanner(System.in);
            int x, y;
            System.out.println("Enter two numbers: ");
            x = sc.nextInt();
            y = sc.nextInt();
            if(x > y)
                  System.out.println(x + " is greater than " + y);
            else
                  System.out.println(x + " is lesser than " + y);
      }
 }
```

Output -
Enter two numbers: 3 2
3 is greater than 2

2. Program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop. (Hint Fact of 4 = 4*3*2*1)

```
public class Factorial{
      public static void main(String args[]){
            int i, n, fact;
            for(i = 1; i <=10; i++)
            {
                  fact = 1;
                  n = i;
                  while(n > 1)
                  {
                        fact *= n;
                        n--;
                  }
                  System.out.println("Factorial of " + i + " = " + fact);
            }
      }
 }
```

Output -

Factorial of 1 = 1
Factorial of 2 = 2
Factorial of 3 = 6
Factorial of 4 = 24
Factorial of 5 = 120
Factorial of 6 = 720
Factorial of 7 = 5040
Factorial of 8 = 40320
Factorial of 9 = 362880
Factorial of 10 = 3628800

3. Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.

```
public class Add{
	void sum(int x, int y)
	{
		int s = x + y;
		System.out.println("Sum = " + s);
	}
	void sum(float x, float y)
	{
		float s = x + y;
		System.out.println("Sum = " + s);
	}
	void sum()
	{
		int x = 5;
		int y = 10;
		int s = x + y;
		System.out.println("Sum = " + s);
	}
	public static void main(String args[])
	{
		Add a = new Add();
		a.sum(1, 2);
		a.sum(1.1f, 2.2f);
		a.sum();
	}
 }
```

Output -
Sum = 3
Sum = 3.3000002
Sum = 15

4. Program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the Superclass. MulDiv should have methods to multiply and divide .A main function should access the methods and perform the mathematical operations.

```
class AddSub{
	void add(int x, int y){
			int s = x + y;
			System.out.println("Sum = " + s);
	}

	void sub(int x, int y){
			int s = x - y;
			System.out.println("Difference = " + s);
	}
}

class MulDiv extends AddSub{
	void mul(int x, int y){
			int m = x * y;
			System.out.println("Product = " + m);
	}

	void div( int x, int y){
			int d = x / y;
			System.out.println("Quotient = " + d);
	}

	public static void main(String args[]){
			MulDiv md = new MulDiv();
			md.add(20, 10);
			md.sub(20, 10);
			md.mul(20, 10);
			md.div(20, 10);
	}

}
```

Output -

Sum = 30
Difference = 10
Product = 200
Quotient = 2

5. Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.

```
class Student{
	int roll;
	String name;
	static String college;

	Student(int r, String n, String c){
		roll = r;
		name = n;
		college = c;
	}
	void display()
	{
		System.out.println(roll + " " + name + " " + college);
	}
 }
class StaticExample{
	public static void main(String args[]){
		Student s1 = new Student(1, "s1", "sample1");
		s1.display();
		Student s2 = new Student(2, "s2", "sample2");
		s2.display();
		s1.display();
	}
}
```

Output-

1 s1 sample1
2 s2 sample2
1 s1 sample2

6. Program

a) To find the area and circumference of the circle by accepting the radius from the user.

```
import java.util.Scanner;
class A06aCircle{
	public static void main(String args[]){
		Scanner sc = new Scanner(System.in);
		System.out.print("Enter the radius: ");
		float r = sc.nextFloat();
		float a = 3.14f * r * r;
		float c = 2 * 3.14f * r;
		System.out.println("Area = " + a);
		System.out.println("Circumference = " + c);
	}
}
```

Output-

Enter the radius: 3
Area = 28.26
Circumference = 18.84

b) To accept a number and find whether the number is Prime or not.

```
import java.util.Scanner;
class A06bPrime{
	public static void main(String[] args){
		Scanner sc = new Scanner(System.in);
		int i, c = 0;
		System.out.print("Enter the number: ");
		int n = sc.nextInt();
		if(n <= 1)
			System.out.println("It is not a prime number");
		else
		{
			for ( i = 2; i < n; i++)
			{
				if (n % i == 0)
					c++;
			}
			if (c == 0)
			System.out.println("It is a prime number.");
			else
				System.out.println("It is not a prime number.");
		}
	}
}
```

Output -

Enter the number: 3

It is a prime number.

7. Program to create a student class with following attributes; Enrollment No: Name, Mark of sub1, Mark of sub2, mark of sub3, Total Marks. Total of the three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50. If a candidate fails in any one of the subjects his total mark must be declared as zero. Using this condition, write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.

```
import java.util.Scanner;
public class A07Constructor{
            public static void main(String args[]){
                        StudentInfo stu[] = new StudentInfo[3];
                        for(int i = 0; i < 3; i++)
                                    stu[i] = new StudentInfo();
                        System.out.println("\t\tStudent Details");
                        System.out.println("Enrollment
no.\t\tName\t\tTotal");
                        for(int i = 0; i < 3; i++)
                                    stu[i].displayInfo();
            }
}
class StudentInfo{
            Scanner sc = new Scanner(System.in);
            String eid;
            String name;
            int s1, s2, s3, total;
            StudentInfo(){
                        readStudentInfo();
            }
            public void readStudentInfo(){
                  System.out.println("Enter student details");
                  System.out.println("Enrollment no: ");
                  eid = sc.next();
                  System.out.println("Name: ");
                  name = sc.next();
                  System.out.println("Enter marks of 3 subjects: ");
                  s1 = sc.nextInt();
                  s2 = sc.nextInt();
                  s3 = sc.nextInt();
                  if(s1 >= 50 && s2 >= 50 && s3 >= 50)
                        total = s1 + s2 + s3;
                  else
```

```
                        total = 0;
            }

            public void displayInfo(){
                  System.out.println(eid + "\t\t\t" + name + "\t\t\t" +
total);
            }
}
```

Output -

Enter student details
Enrollment no: 1
Name: sample1
Enter marks of 3 subjects: 99 99 99
Enter student details
Enrollment no: 2
Name: sample2
Enter marks of 3 subjects: 1 1 1
Enter student details
Enrollment no: 3
Name: sample3
Enter marks of 3 subjects: 12 5 1

Student Details

Enrollment no.	Name	Total
1	sample1	297
2	sample2	0
3	sample3	0

8. In a college first year class are having the following attributes

1. Name of the class (BCA, BCom, BSc),
2. Name of the staff (Class Teacher Name)
3. No of the students in the class
4. Array of students in the class
5. Display Student Information

```
import java.util.Scanner;
public class FirstYearClassRoom {
	String className;
	String classTeacherName;
	int studentCount;
	String studentName[] = new String[50];  // Student Name Array
	Scanner sc =new Scanner(System.in);
   // Below is constructor function that is called when you create object
of this class
	public FirstYearClassRoom(){
		getInfo();
	}


	private void getInfo(){
		System.out.print("Enter the class name:");
		className = sc.nextLine();

		System.out.print("Enter class teacher name: ");
		classTeacherName = sc.nextLine();

		System.out.print("Enter total number of students in the class:
");
		studentCount = Integer.parseInt(sc.nextLine());

		System.out.println("Enter names of " + studentCount + "
students:");
		for(int i = 0; i < studentCount; i++)
			studentName[i] = sc.nextLine();
	}

	public void display(){
		System.out.println("\nClass Name: " + className);
		System.out.println("Class Teacher Name: " + classTeacherName);
		System.out.println("\t\t\tStudent Names ");
		System.out.println("\t\t------------------------ ");
```

```
            for(int i = 0; i < studentCount; i++)
                  System.out.println(studentName[i]);
      }

      public static void main (String args[]){
            FirstYearClassRoom  fy = new FirstYearClassRoom();  // this
will trigger getinfo function
            fy.display();
      }
}
```

Output -

Enter the class name:BCA
Enter class teacher name: Teacher1
Enter total number of students in the class: 3
Enter names of 3 students: student1
student2
student3
Class Name: BCA
Class Teacher Name: Teacher1
Student Names
---
student1
student2
student3

9. Define a class called first year with above attributes and define a suitable constructor. Also write a method called best Student () which processes a first-year object and returns the student with the highest total mark. In the main method, define a first-year object and find the best student of this class.

```
import java.util.Scanner;
public class A09FirstYearClassBestMarks {

    String className;
    String  classTeacherName;
    int studentCount;
    String studentName[] = new String[50];  // Student Name Array
    int studentMarks[ ] = new int[50];    // Student Marks Array
    Scanner sc  = new Scanner(System.in);

    // Below is constructor function that is called when you create object
of this class
    public A09FirstYearClassBestMarks(){
       getInfo();
    }

    private void getInfo(){

       System.out.print("Enter the class name: ");
       className = sc.nextLine();

       System.out.print("Enter class teacher name: ");
       classTeacherName = sc.nextLine();

       System.out.print("Enter number of students: ");
       studentCount = Integer.parseInt(sc.nextLine());

       System.out.println("Please enter names " + studentCount + "
students: ");
       for(int i = 0; i < studentCount; i++)
                   studentName[i] = sc.nextLine();

       System.out.println("Enter marks for students ");
       for(int i = 0; i < studentCount; i++){
             System.out.print("Enter marks for " + studentName[i] + ": ");
             studentMarks[i] =sc.nextInt();
             System.out.println();  // go to next line
       }
```

```
    }

    public void display(){
       System.out.println("\n\nClass and Student Info");
       System.out.println("Class Name: " + className);
       System.out.println("Class Teacher Name: " + classTeacherName);
       System.out.println("Student Names\t\tMarks");
       System.out.println("-------------\t\t------");
       for(int i = 0; i < studentCount; i++)
             System.out.println(studentName[i] + "\t\t\t" +
studentMarks[i]);
    }

    public void bestStudent(){
             int best = 0;  // variable to keep track of best marks till
now when looping
             int k = -1 ;   // variable to keep track of best
student-index. Note that since index start from 0, initialize this to -1.

             // loop through all student marks and pick best marks.
             for(int i= 0; i <= studentCount; i++){
                        if(studentMarks[i] > best){
                                   best = studentMarks[i];
                                   k = i;  // keep track of index. As
student at this index is best marks.
                        }
             }

                   // once out of loop you will have student who got best
marks. student index is k.
             System.out.println("The best student is " + studentName[k]);
             System.out.println("They scored marks which was highest - " +
studentMarks[k]);
       }

    public static void main (String args[]){
       A09FirstYearClassBestMarks fy = new A09FirstYearClassBestMarks();
// this will trigger getinfo function
       fy.display();
       fy.bestStudent();
    }
}
```

Output -

```
Enter the class name: BCA
Enter class teacher name: Teacher1
Enter number of students: 3
Please enter names 3 students:
student1
student2
student3
Enter marks for students:
Enter marks for student1: 99

Enter marks for student2: 95

Enter marks for student3: 12



Class and Student Info
Class Name: BCA
Class Teacher Name: Teacher1
Student Names          Marks
-------------          ------
student1                    99
student2                    95
student3                    12
The best student is -student1
They scored marks which was highest - 99
```

10. Program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment. ie, print them as per their seniority.

```
import java.util.Date;

public class Employee {
    String employeeName;
    Date appointmentDate;

    //Constructor
    public Employee(String name, Date aptDate){
       employeeName = name;
       appointmentDate = aptDate;
    }

    public static void main (String args[]){
       Employee emp[] = new Employee[10];  // array of 10 employee objects

       emp[0] = new Employee("Ritika", new Date(2000,1,25));
       emp[1] = new Employee("Rohit", new Date(1999,1,25));
       emp[2] = new Employee("Himesh", new Date(2010,1,25));
       emp[3] = new Employee("Gajendra", new Date(2022,1,15));
       emp[4] = new Employee("Shilpa", new Date(2008,1,25));
       emp[5] = new Employee("Deepa", new Date(2001,1,25));
       emp[6] = new Employee("Nandini", new Date(2012,1,25));
       emp[7] = new Employee("Ramesh", new Date(2010,1,28));
       emp[8] = new Employee("Naseer", new Date(2011,1,25));
       emp[9] = new Employee("Swetha", new Date(1990,1,25));

       System.out.println("Display Employee List");
       for(int i=0 ; i < emp.length; i++)
                  System.out.println("Employee Name: " +
emp[i].employeeName + " Appointment date: " +
emp[i].appointmentDate.getDate() + "/" + emp[i].appointmentDate.getMonth()
+ "/" + emp[i].appointmentDate.getYear());

       // Do sorting. Sort based on date.
       for(int i=0 ; i < emp.length; i++) {
             for (int j = i + 1; j < emp.length; j++) {
                   if(emp[i].appointmentDate.after(emp[j].appointmentDate))
{
                         //swap
```

```
                        Employee temp = emp[i];
                        emp[i] = emp[j];
                        emp[j] = temp;
                    }
                }
        }
        System.out.println("----------------------------------------");
        System.out.println("Display Sorted Employee List as per Seniority
");
        for(int i = 0 ; i < emp.length; i++){
              System.out.println("Employee Name: " + emp[i].employeeName + "
Appointment date: " + emp[i].appointmentDate.getDate() + "/" +
emp[i].appointmentDate.getMonth() + "/" + emp[i].appointmentDate.getYear()
);
        }
    }
}
```

Output -

Display Employee List
Employee Name: Ritika Appointment date: 25/1/2000
Employee Name: Rohit Appointment date: 25/1/1999
Employee Name: Himesh Appointment date: 25/1/2010
Employee Name: Gajendra Appointment date: 15/1/2022
Employee Name: Shilpa Appointment date: 25/1/2008
Employee Name: Deepa Appointment date: 25/1/2001
Employee Name: Nandini Appointment date: 25/1/2012
Employee Name: Ramesh Appointment date: 28/1/2010
Employee Name: Naseer Appointment date: 25/1/2011
Employee Name: Swetha Appointment date: 25/1/1990

Display Sorted Employee List as per Seniority
Employee Name: Swetha Appointment date: 25/1/1990
Employee Name: Rohit Appointment date: 25/1/1999
Employee Name: Ritika Appointment date: 25/1/2000
Employee Name: Deepa Appointment date: 25/1/2001
Employee Name: Shilpa Appointment date: 25/1/2008
Employee Name: Himesh Appointment date: 25/1/2010
Employee Name: Ramesh Appointment date: 28/1/2010
Employee Name: Naseer Appointment date: 25/1/2011
Employee Name: Nandini Appointment date: 25/1/2012
Employee Name: Gajendra Appointment date: 15/1/2022

Part B

1. Program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.

```
class B01NegativeSize{
	public static void main(String args[]){
		try{
			int[] a = new int[-5];
		}
		catch(NegativeArraySizeException e){
			System.out.println("Negative Array Size!!");
		}
		System.out.println("Continuing execution... ");
	}
}
```

Output -

Negative Array Size!!
Continuing execution…

2. Program to handle Null Pointer Exception and use the “finally” method to display a message to the user.

```
public class B02NullPointer{
	public static void main(String args[]){
		String str = null;
		//Try-catch-finally blocks
		try{
			// Below code will give null pointer exception as we are
access null str
			// Note that string is a character array. And we are
trying to access the 1st character of this string.
			System.out.println("First character of str is:" +
str.charAt(0));
	}
	catch(NullPointerException e){
		System.out.println("NullPointerException Caught in catch
block...");
	}
	finally{
		System.out.println("Finally is executed always");
		}
	}
}
```

Output-

NullPointerException Caught in catch block…
Finally is executed always

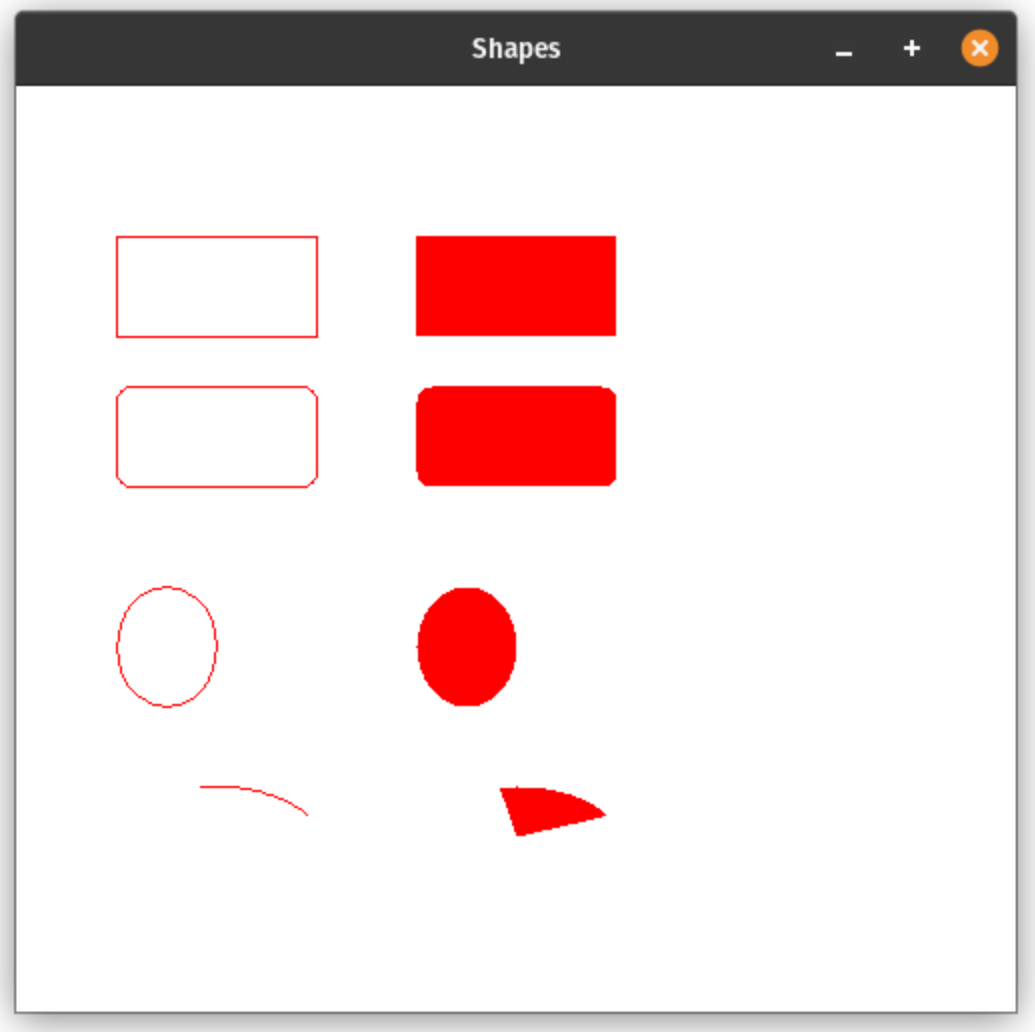
3. Program which creates and displays a message on the window.

```
import java.applet.*;
import java.awt.*;
public class AppletWindow extends Applet{
      public void paint(Graphics g){
            g.drawString(“Hello World”, 30, 30);
      }
}
```

Output -

4. Program to draw several shapes in the created window

```
import java.awt.*;

public class DrawShapes extends Canvas {
	public void paint (Graphics g)
	{
		// Rectangle shape
		setBackground(Color.WHITE);
		setForeground(Color.RED);
		g.drawRect(50,75,100,50);
		g.fillRect(200,75,100,50);

		// Rounded corners Rectangle shape
		g.drawRoundRect(50,150,100,50,15,15);
		g.fillRoundRect(200,150,100,50,15,15);

		//Oval shape
		g.drawOval(50,250,50, 60);
		g.fillOval(200,250,50, 60);

		// Arch - with changed background and foreground
		g.drawArc(50, 350, 100,50,25,75);
		g.fillArc(200, 350, 100,50,25,75);
	}

	public static void main (String args[]){
		DrawShapes m = new DrawShapes();
		Frame f = new Frame("Shapes");
		f.add(m);
		f.setSize(500, 500);
		f.setVisible(true);
	}
}
```


Shapes

8. Create a simple applet which reveals the personal information of yours

```
import java.awt.*;
import java.applet.*;
public class Personal extends Applet{
	Font f;
	public void init(){
		setSize(500,500);
		f=new Font ("Arial",Font.BOLD,20);
	}
	public void paint(Graphics g){
		g.drawString("==============", 50, 50);
		g.drawString("Personal Information", 50, 80);
		g.drawString("==============", 50, 110);
		g.drawString("Name:Charles", 50, 140);
		g.drawString("Father Name:James", 50, 170);
		g.drawString("Date of Birth:02/03/1995", 50, 200);
		g.drawString("Address No:50,MG Road,Kolar", 50, 230);
		g.drawString("Mobile Number:9123456789", 50, 260);
		g.drawString("E-mail:charles1995@gmail.com", 50, 290);
	}
}
/* <applet code=”Personal.class” height=300 width=500> </applet> */
```

Output -

11. Demonstrate the various mouse handling events using suitable example.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class B11MouseEvents extends Applet implements MouseListener{
	String msg = "Initial Message";
	public void init(){
		addMouseListener(this);
	}
	public void mouseClicked(MouseEvent m){
		msg = "Mouse Clicked";
		repaint();
	}
	public void mousePressed(MouseEvent m){
		msg = "Mouse Pressed";
		repaint();
	}
	public void mouseReleased(MouseEvent m){
		msg = "Mouse Released";
		repaint();
	}
	public void mouseEntered(MouseEvent m){
		msg = "Mouse Entered";
		repaint();
	}
	public void mouseExited(MouseEvent m){
		msg = "Mouse Exited";
		repaint();
	}
	public void paint(Graphics g){
		g.drawString(msg, 30, 30);
	}
}
```

Output -