**1A.Create class and implement a default overloaded and copy constructor.**

```
class MyClass
{
	private int x;
	public MyClass()
	{
		System.out.println("Default Constructor");
	}
	public MyClass(int a)
	{
		x=a;
		System.out.println("Parameterised constructor and 
value is "+x);
	}
	public MyClass(MyClass b)
	{
		x=b.x;
		System.out.println("Copy constructor and value is "+x);
	}
}
class Pract1a
{
	public static void main(String args[])
	{
		MyClass obj1=new MyClass();
		MyClass obj2=new MyClass(7);
		MyClass obj3=new MyClass(obj2);
	}
}
```

**1B.Create a class and implement the concepts of method overloading.**

```
class MyClassOver
{
	public int add(int a,int b)
	{
		return a+b;
	}
	public int add(int a,int b,int c)
	{
		return a+b+c;
	}

}
class Pract1b
{
	public static void main(String args[])
	{
		MyClassOver obj=new MyClassOver();
		int sum1=obj.add(10,20);
		int sum2=obj.add(10,20,30);
		System.out.println("Sum of two int "+sum1);
		System.out.println("Sum of three int "+sum2);
	}
}
```

**1C.create a class and implement the concepts of static method.**

```
class StaticMethod
{
	public static int add(int a,int b)
	{
		return a+b;
	}
	public static int sub(int a,int b)
	{
		return a-b;
	}
}
class Pract1c
{
	public static void main(String args[])
	{
		int s=StaticMethod.add(8,4);
		int d=StaticMethod.sub(8,4);
		System.out.println("Sum= "+s);
		System.out.println("Sum= "+d);
	}
}
```

**2A.implement the concepts of inheritance and method overriding.**

```
class A
{
	void show()
	{
		System.out.println("Base Class");
	}
}
class B extends A
{
	void show()
	{
		System.out.println("Derived Class");
	}
}
class Pract2a
{
	public static void main(String args[])
	{
		B b=new B();
		b.show();
	}
}
```

**2B.implement the concept of abstract classes and methods.**

```
abstract class Shape{
	public abstract double area();
}
class Circle extends Shape{
	private double r;
	public Circle(double r){
		this.r=r;
	}
	public double area(){
		return Math.PI*r*r;
	}
}
public class Pract2b{
	public static void main(String args[]){
		Circle c=new Circle(10.0);
		System.out.println("Area of circle:
"+c.area());
	}
}
```

**2C.implement the concept of inheritance.**

```
interface Shape{
	double area();
	double perimeter();
}
class Circle implements Shape{
	private double r;
	public Circle(double r){
		this.r=r;
	}
	public double area(){
		return Math.PI*r*r;
	}
	public double perimeter(){
		return 2*Math.PI*r;
	}
}
class Pract2c{
	public static void main(String args[]){
		Circle c=new Circle(10.0);
		System.out.println("Area of circle
"+c.area());
		System.out.println("Perimeter of circle
"+c.perimeter());  }  }
```

**3B.user define exceptions and raise them as per the requirement.**

```
class CustomException extends Exception{
    public CustomException(String msg){
        super(msg);
    }
}
class Pract3b{
    public static void main(String args[]){
        try{
            int age=20;
            if(age<0){
                throw new
CustomException("Age cannot be negative!");
            }
            System.out.println("age="+age);
        }
        catch(CustomException e){
            System.out.println("Error
"+e.getMessage());
        }
    }  }
```

**7A.FlowLayout**

```
import javax.swing.*;
import java.awt.*;
public class DemoFlowLayout
{
	public static void main(String args[])
	{
		JFrame f=new JFrame("Flow Layout
Demo");
		f.setSize(100,100);
	f.setDefaultCloseOperation(JFrame.EXIT_ON_
CLOSE);
		JPanel pan = new JPanel(new
FlowLayout(FlowLayout.RIGHT));
		JButton btn1= new JButton("Button-1");
		JButton btn2= new JButton("Button-2");
		JButton btn3= new JButton("Button-3");
		pan.add(btn1);
		pan.add(btn2);
		pan.add(btn3);
		f.add(pan);
		f.setVisible(true);
	}
}
```

**7B.GridLayout**

```
import javax.swing.*;
import java.awt.*;
public class DemoGridLayout
{
	public static void main(String agrs[])
	{
		JFrame f=new JFrame("Flow Grid Demo");
		f.setSize(300,200);
	f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOS
E);
		JPanel pan=new JPanel(new GridLayout(2,3));
		JButton btn1= new JButton("Button-1");
		JButton btn2= new JButton("Button-2");
		JButton btn3= new JButton("Button-3");
		JButton btn4= new JButton("Button-4");
		JButton btn5= new JButton("Button-5");
		JButton btn6= new JButton("Button-6");
		pan.add(btn1);
		pan.add(btn2);
		pan.add(btn3);
		pan.add(btn4);
		pan.add(btn5);
		pan.add(btn6);
		f.add(pan);
		f.setVisible(true);   }    }
```

**7C.BordelLayout**

```
import javax.swing.*;
import java.awt.*;
public class DemoBorderLayout {
	public static void main(String args[]) {
		JFrame f= new JFrame("Border Layout
Demo");
		f.setSize(300,200);
	f.setDefaultCloseOperation(JFrame.EXIT_ON_CL
OSE);
		JButton btn1= new JButton("North");
		JButton btn2= new JButton("South");
		JButton btn3= new JButton("West");
		JButton btn4= new JButton("East");
		JButton btn5= new JButton("Center");
		Container contentpane=f.getContentPane();
		contentpane.setLayout(new
BorderLayout());
	contentpane.add(btn1,BorderLayout.NORTH);
	contentpane.add(btn2,BorderLayout.SOUTH);
	contentpane.add(btn3,BorderLayout.WEST);
	contentpane.add(btn4,BorderLayout.EAST);
	contentpane.add(btn5,BorderLayout.CENTER);
		f.setVisible(true);
	}   }
```

**4.Bouncing ball**

```
import java.awt.*;
import java.awt.event.*;
class Bouncing_Balls extends Frame implements MouseListener
{
	int x=40, y=40,t1=1,t2=1;
	int x1=200, y1=40,t12=1,t22=1;
	int x2=100, y2=100,t13=1,t23=1;
	int x3=300, y3=40,t14=1,t24=1;
	int x4=400, y4=100,t15=1,t25=1;
	Thread th;
	Bouncing_Balls(){
		setSize(500,500);
		setVisible(true);
		th=new Thread(new Thread(){
		public void run() {
			while(true){
x=x+t1;
	y=y+t2;x1=x1+t12;=y1+t22;x2=x2+t13;y2=y2+t23;x3=x3+t14;y3
=y3+t24;x4=x4+t15;y4=y2+t25;

					if(x<0||x>500)
						t1=t1*(-1);
					if(y<20||y>500)
						t2=t2*(-1);
					---------
					try{
					this.sleep(5);
				}
					catch(Exception E){
				}

					repaint();
```

**4.**

```
            }
          }
    });
        addMouseListener(this);
}
        public void
        mouseClicked(MouseEvent M)
   {
        th.start();
   }
        public void mousePressed(MouseEvent M){}
        public void mouseReleased(MouseEvent M){}
        public void mouseEntered(MouseEvent M){}
        public void mouseExited(MouseEvent M){}
        public void paint (Graphics g)
  {
                g.setColor(Color.green);
                g.fillOval(x,y,40,40);
               g.setColor(Color.yellow);
                g.fillOval(x1,y1,40,40);
                g.setColor(Color.pink);
                g.fillOval(x2,y2,40,40);
                g.setColor(Color.blue);
                g.fillOval(x3,y3,40,40);
                g.setColor(Color.black);
                g.fillOval(x4,y4,40,40);
    }
        public static void main (String args[])
    {

        Bouncing_Balls B=new Bouncing_Balls ();
 }
}
```

6A.swing application that randomly changes color on bottom click

```
import javax.swing.*;
import java.awt.*;
import java.util.Random;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class Changedcolor extends JFrame {
	private JPanel colorPanel;
	private JButton changeColorButton;
	public Changedcolor() {
		setTitle("Random color change");
		setSize(300,200);

	setDefaultCloseOperation(JFrame.EXIT_ON_C
LOSE);
		setLayout(new BorderLayout());
		colorPanel = new JPanel();
		changeColorButton = new
JButton("Change Color");
		add(colorPanel, BorderLayout.CENTER);
		add(changeColorButton,
BorderLayout.SOUTH);
```

```
        changeColorButton.addActionListener(new
ActionListener() {
                public void
actionPerformed(ActionEvent e) {
                    ChangeColor();
                }
            });
        }
        public void ChangeColor() {
            Random random = new Random();
6A.         Color randomColor = new
Color(random.nextInt(256), random.nextInt(256),
random.nextInt(256));
            colorPanel.setBackground(randomColor);
        }
        public static void main(String args[]) {
            SwingUtilities.invokeLater(()->{
                Changedcolor app = new
Changedcolor();
                app.setVisible(true);
            });
        }
}
```

**6C.scrollpane to change color**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ColorChanger extends JFrame {
	private JScrollPane jscroll = new JScrollPane();

	private JButton colorchangebtn = new
JButton("Change Color");
	public ColorChanger () {
		setTitle("ScrollPane color changer");
		setSize(400, 300);
	setDefaultCloseOperation(JFrame.EXIT_ON_C
LOSE);
		setLayout(new BorderLayout());
		add(jscroll, BorderLayout.CENTER);
		add(colorchangebtn,
BorderLayout.SOUTH);
		colorchangebtn.addActionListener(new
ActionListener() {
			public void
actionPerformed(ActionEvent e) {
```

**6C.**

```
            Color
selectColor=JColorChooser.showDialog
                        (null, " choose a color",
jscroll.getBackground());
                        if (selectColor != null) {
      jscroll.getViewport().setBackground(selectCol
or);
                    }
            }
      });
}
public static void main(String agrs[])
{
      new ColorChanger().setVisible(true);
}
}
```

# 6B.filechooser

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.FileReader;
public class showFile extends JFrame {
	private JTextArea txtArea = new JTextArea(20,40);
	private JButton openFilebtn= new JButton("Open File");
	public showFile() {
		setTitle("File viewer");
		setSize(300,200);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setLayout(new BorderLayout());
		JScrollPane scrollpane = new JScrollPane(txtArea);
		add(scrollpane,BorderLayout.CENTER);
		add(openFilebtn, BorderLayout.SOUTH);
		openFilebtn.addActionListener( new
ActionListener() {
			public void actionPerformed(ActionEvent e) {
				openFile();
			}
		});
	}
	public void openFile() {
		JFileChooser filechooser= new JFileChooser();
```

```
            int result= filechooser.showOpenDialog(this);

            if(result==JFileChooser.APPROVE_OPTION) {
                try(BufferedReader reader=new
BufferedReader(
                        new
FileReader(filechooser.getSelectedFile()))) {
                        StringBuilder content = new
StringBuilder();
                        String line;
                        while((line=reader.readLine()) != null) {
                              content.append(line).append("/n");
                        }
                        txtArea.setLineWrap(true);
                        txtArea.setText(content.toString());
                }
                catch(IOException e) {
                JOptionPane.showMessageDialog(this,"Error
reading the file","Error",JOptionPane.ERROR_MESSAGE);
                }
            }
      }
      public static void main(String args[]) {
            new showFile().setVisible(true);
      }
}
```