1 wap to create class and implement a default overload

```
class Q1
{
	private int data;

	//Default Constructor
	public Q1()
	{
		this.data = 10;
	}

	// Overloaded Constructor
	public Q1(int data)
	{
		this.data = data;
	}

	// Copy Construct
	public Q1(Q1 value)
	{
			this.data = value.data;
	}

	// Getter method to retrieve the value
	public int GetData()
```

```
	{
		return data;
	}

	public static void main(String[] args)
	{
		// Using the default constructor
		Q1 obj1 = new Q1();
		System.out.println("\nDefault Constructor - Data for obj1 : " + obj1.GetData());

		// Using the overloaded constructor
		Q1 obj2 = new Q1(100);
		System.out.println("\nOverloaded Constructor - Data for obj2 : " + obj2.GetData());

		// Using the copy constructor
		Q1 obj3 = new Q1(obj2);
		System.out.println("\nCopy Constructor - Data for obj3 : " + obj3.GetData());
	}//main()
}//class Q1
```

2 wap to create class implement concept of method overloadinhg

```
class Q2
{
	void Show()
	{
		System.out.println("Show() method - 1");
```

```
	}//Show()


	void Show(int a)
	{
		System.out.println("Show() method - 2");
	}//Show()


	public static void main(String [] args)
	{
		Q2 m=new Q2();
		m.Show(1000);
	}//main()
}//class Q2
```

3 wap to class static method

```
import java.util.*;
class Q3
{
	//Static method
	public static int Addition(int x, int y)
	{
			return x + y;
	}//Addition


	//Static method
	public static int Subtraction(int x, int y)
```

```
{
        return x - y;
}//Subtraction

//Non-Static method
public int Multiplication(int x, int y)
{
        return x * y;
}//Multiplication

//Non-Static method
public int Divison(int x, int y)
{
        return x / y;
}//Divison

public static void main(String [] args)
{
        int add, sub, mul, div, num1, num2;

        Scanner sc=new Scanner(System.in);

        System.out.println("------Data for Static and Non-Static Methods------");
        System.out.print("Enter num1 : ");
        num1=sc.nextInt();
```

```
        System.out.print("\nEnter num2 : ");
        num2=sc.nextInt();

        System.out.println("\n------Calling a Static Methods------");
        //Calling a static method
        add=Addition(num1,num2);
        System.out.println("Addition : "+ add);

        //Calling a static method
        sub=Subtraction(num1,num2);
        System.out.println("\nSubtraction : "+ sub);

        //Object creation to call non-static methods
        Q3 obj=new Q3();

        System.out.println("\n------Calling a Non-Static Methods------");
        //Calling a non-static method
        mul=obj.Multiplication(num1,num2);
        System.out.println("Multiplication : "+ mul);

        //Calling a non-static method
        div=obj.Divison(num1,num2);
        System.out.println("\nDivison : "+ div);
}//main()
```

```
}//class Q3
```

4 wap to inheritance and method overriding

```
class A
{
	void Show()
	{
		System.out.println("Show() - 1");
	}//Show()
}//class A

class B extends A
{
	void Show()
	{
		//super.Show(); //When you want to call parent class method
		System.out.println("Show() - 2");
	}//Show()
}//class B

class Q4
{
	public static void main(String [] args)
	{
		B obj=new B();
```

```
                obj.Show();
        }//main()
}//class Q4
5
abstract class Vehicle //Vehicle is Parent Class
{
        abstract void Start();
}//class Vehicle


//Inheritance
class Car extends Vehicle //Car is Child Class
{
        void Start()
        {
                System.out.println("Car starts with a KEY");
        }//Start()
}//class Car
class Bike extends Vehicle //Bike is Child Class
{
        void Start()
        {
                System.out.println("Bike starts with a KICK");
        }//Start()
        public static void main(String [] args)
        {
```

```
                Car c=new Car();
                c.Start();

                Bike b=new Bike();
                b.Start();
        }//main()
}//class Bike
```

```
//Save the file - Vehicle.java

//Compile - javac Vehicle.java

//Execute - java Bike
```

6

cocept of interface

```
interface Bank
{
        float ROI();
}//interface Bank
interface SavingsAccount
{
        void Account();
}//interface SavingsAccount
class LenaBank implements Bank, SavingsAccount
{
```

```
	public float ROI()
	{
		return 8.2F;
	}//ROI()
	public void Account()
	{
		System.out.println("\nSavings Account");
	}//Account()
}//class LenaBank
public class Q6
{
	public static void main(String [] args)
	{
		Bank b1=new LenaBank();
		System.out.println("LenaBank - "+b1.ROI());

		SavingsAccount a1=new LenaBank();
		a1.Account();
	}//main()
}//class Q6
```

7 one D array

```
class Q7
{
	public static void main(String [] args)
	{
```

```
        int[] a={5,10};
        int b=5;

        try
        {
                int x=a[2]/b-a[1];
        }//try

        catch(ArithmeticException e)
        {
                System.out.println("Division by Zero");
        }//catch
        catch(IllegalArgumentException e)
        {
                System.out.println("Class not found");
        }//catch
        catch(ArrayIndexOutOfBoundsException e)
        {
                System.out.println("Array index error");
        }//catch
        finally
        {
                System.out.println("Application Executed");
        }//finally
}//main()
```

```
}//class Q7
```

8 wap to user define exception amd rause them as the reqirements

```
import java.util.*;

class NumberNotInRange extends Exception

{

	NumberNotInRange(String str)

	{

		//Call the constructor of parent class Exception

		super(str);

	}//NumberNotInRange

}//class NumberNotInRange


class Q8

{

	public static void main(String [] args)

	{

		int num;

		Scanner sc=new Scanner(System.in);


		System.out.println("Enter the number:");

		num=sc.nextInt();


		try

		{

			if(num<=100 && num>=1)
```

```
                {
                        System.out.println("Your inputs are correct");
                }//if
                else
                {
                        throw new NumberNotInRange("The number should be in the range of
1 to 100");
                }//else
        }//try
        catch(NumberNotInRange e)
        {
                System.out.println(e.getMessage());
        }//catch
    }//main()
}//class Q8
```

9

wap data hiding and encapsulation

```
class emp
{
    private int empid;
    public void setempid(int eid)
    {
        empid=eid;
    }
    public int getempid()
```

```
	{
		return empid;
	}
}
class Q9
{
	public static void main(String [] args)
	{
		emp e=new emp();
		e.setempid(007);
		System.out.println(e.getempid());
	}
}
```

10

wap to demonstrate arithmetic exceptions

```
class Q10
{
	public static void main(String [] args)
	{
		int a,b,c;
		java.util.Scanner sc=new java.util.Scanner(System.in);
		System.out.println("enter a:");
		a=sc.nextInt();
		System.out.println("enter b:");
```

```
                b=sc.nextInt();
                System.out.println("\na="+a);
                System.out.println("\nb="+b);
                try
                {
                        c=a/b;
                        System.out.println("\nc="+c);
                }
                catch(ArithmeticException e)
                {
                        System.out.println("pleas enter non zero number");
                }

        }
}
```

11 wap multiple cathch blocks along with family

```
public class Q11
{
        public static void main(String [] args)
        {
                try
                {
                        int[] num={1,2,3};
                        int result= num[4];
```

```
                System.out.println("result="+result);
            }
            catch(ArrayIndexOutOfBoundsException e)
            {
            System.out.println("ArrayIndexOutOfBoundsException");
            }
            finally
            {
                System.out.println("finally block executed");
            }
        }
}
```

12 wap design thread by exrenduga thread class

```
class mythread extends Thread
{
    public void run()
    {
        for(int i=0;i<=5;i++)
        {
            System.out.println("my thread class");
        }
    }
}
class Q12
{
```

```
	public static void main(String [] args)


	{
		mythread t=new mythread();
		t.start();
		for(int i=0;i<=5;i++)
		{
			System.out.println("Q12 class");
		}
	}


}
```

13

14

awt extending frames class

```
import java.awt.*;

class Q14 extends Frame

{
	Q14()
	{
		//Button class
		Button b=new Button("Click Me");


		//Setting button position

/*setBounds(int x, int y, int width, int height) - Specifies the size of the frame and the location of the upper left corner x axis to the right and y axis to the bottom*/

		b.setBounds(30,100,80,30);
```

```
		//Adding button into frame
		add(b);

		//Frame size - 300 width and 300 height
		setSize(300,300);

		//No layout manager
		setLayout(null);

		//Now frame will be visible, by default not visible
		setVisible(true);
	}
	public static void main(String args[])
	{
		Q14 p=new Q14();
	}
}
```

15

awt text area

```
import java.awt.*;
class Q15
{
	public static void main(String args[])
	{
			Frame f= new Frame();
```

```
                TextArea t=new TextArea();
                t.setBounds(10,30,300,200);
                f.add(t);
                f.setSize(400,400);
                f.setLayout(null);
                f.setVisible(true);
        }
}
```

16

awt labels and buttons

```
import java.awt.*;
class Q16
{
        public static void main(String args[])
        {
                Frame f= new Frame("Labels and Buttons");

                //Labels
                Label l1,l2;
                l1=new Label("Label 1");
                l1.setBounds(50,100, 100,30);

                l2=new Label("Label 2");
                l2.setBounds(50,150, 100,30);

                f.add(l1);
```

```
                f.add(l2);


                //Buttons
                Button b1, b2;
                b1=new Button("Button 1");
                b1.setBounds(160,100, 100,30);

                b2=new Button("Button 2");
                b2.setBounds(160,150, 100,30);

                f.add(b1);
                f.add(b2);

                f.setSize(400,400);
                f.setLayout(null);
                f.setVisible(true);
        }
}
```

17

awt creating object of frame clas

```
import java.awt.*;
class Q17
{
        Q17()
        {
                //Frame class
```

```
		Frame f=new Frame();

		Button b=new Button("Click Me");

		b.setBounds(30,50,80,30);

		f.add(b);

		f.setSize(300,300);

		f.setLayout(null);

		f.setVisible(true);

	}

	public static void main(String args[])

	{

		Q17 p=new Q17();

	}

}
```

18

awt java flow layout

```
import java.awt.*;


class Q18

{

	public static void main(String[] args)

	{

		Frame f=new Frame();


		Button b1=new Button("1");

		Button b2=new Button("2");

		Button b3=new Button("3");
```

```
		Button b4=new Button("4");
		Button b5=new Button("5");

		f.add(b1);
		f.add(b2);
		f.add(b3);
		f.add(b4);
		f.add(b5);

		  //Setting flow layout of right alignment
		f.setLayout(new FlowLayout(FlowLayout.LEFT));

		f.setSize(300,300);
		f.setVisible(true);
	}
}
```

19

awt grid layout

```
import java.awt.*;


class Q19
{
	public static void main(String[] args)
	{
		Frame f=new Frame();
```

```
Button b1=new Button("1");

Button b2=new Button("2");

Button b3=new Button("3");

Button b4=new Button("4");

Button b5=new Button("5");

Button b6=new Button("6");

        Button b7=new Button("7");

Button b8=new Button("8");

Button b9=new Button("9");


f.add(b1);

f.add(b2);

f.add(b3);

f.add(b4);

f.add(b5);

f.add(b6);

f.add(b7);

f.add(b8);

f.add(b9);


//Setting grid layout of 3 rows and 3 columns

f.setLayout(new GridLayout(3,3));


f.setSize(300,300);

f.setVisible(true);
```

```
        }
}
```

20

wap border layout

```
import java.awt.*;


class Q20
{
        public static void main(String[] args)
        {
                Frame f=new Frame();

                Button b1=new Button("NORTH");
                Button b2=new Button("SOUTH");
                Button b3=new Button("EAST");
                Button b4=new Button("WEST");
                Button b5=new Button("CENTER");

                f.add(b1,BorderLayout.NORTH);
                f.add(b2,BorderLayout.SOUTH);
                f.add(b3,BorderLayout.EAST);
                f.add(b4,BorderLayout.WEST);
                f.add(b5,BorderLayout.CENTER);

                f.setSize(500,500);
                f.setVisible(true);
```

```
        }

    }
```