Question 1:

```
package Practice;
class A extends Thread
{	public void run()
	{	boolean flag=false;
		int c=0,i=0;
		while(c<3)
		{
			if(i%2!=0)
			{	System.out.println(i);
				c++;
			}
			i++;
			if(c==3)
			{	flag=true;
				break;
			}
		}
		if(flag)
		{
			try {
				wait();
			}
			catch(Exception e)
			{}
		}
	}
}
class B extends Thread
{	int n;
	B(int n)
	{
		this.n=n;
	}
	public void run()
	{	int reversed = 0;
	while(n != 0) {
		int digit = n % 10;
		reversed = reversed * 10 + digit;
		n /= 10;
	}
	System.out.println("Reversed Number: " + reversed);
	}
```

```
}
public class Main {
	public static void main(String[] args) {
		A a=new A();
		a.start();
		B b=new B(1234);
		b.start();
	}
}
```

Question 2:

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
/*<applet code="Calc" width="500" height="600">
 </applet>*/
public class Calc extends Applet implements ActionListener
{
	Label l1, l2, l3;
	TextField t1,t2,t3;
	Button add1,sub,mul,div;
	public void init()
	{
		l1=new Label("Enter 1st Number : ");
		add(l1);
		t1=new TextField(10);
		add(t1);
		l2=new Label(" Enter 2nd Number : ");
		add(l2);
		t2=new TextField(10);
		add(t2);
		l3=new Label(" Result : ");
		add(l3);
		t3=new TextField(10);
		add(t3);
		add1 = new Button("+");
		add(add1);
		add1.addActionListener(this);
		sub = new Button("-");
		add(sub);
		sub.addActionListener(this);
		mul = new Button("*");
		add(mul);
```

```
		mul.addActionListener(this);
		div = new Button("/");
		add(div);
		div.addActionListener(this);
	}
	public void actionPerformed(ActionEvent ae)
	{
		int res = 0;
		String action = ae.getActionCommand();
		if(action.equals("+")){
		res =Integer.parseInt(t1.getText()) +
		Integer.parseInt(t2.getText());
		t3.setText(String.valueOf(res));
		}
		else if(action.equals("-")){
		res=(Integer.parseInt(t1.getText()) - Integer.parseInt(t2.getText()));
		t3.setText(String.valueOf(res));
		}
		else if(action.equals("*")){
		res=Integer.parseInt(t1.getText()) * Integer.parseInt(t2.getText());
		t3.setText(String.valueOf(res));
		}
		else{
		if(Integer.parseInt(t2.getText()) != 0) {
			res=Integer.parseInt(t1.getText()) /
			Integer.parseInt(t2.getText());
		t3.setText(String.valueOf(res));
		}
		else
			System.out.println("Division can't be performed");
		}
	}
}
```

Question 3:

```
import java.applet.Applet;
import java.awt.event.*;
import java.awt.*;
/*<applet code="Reverse" width="300" height="300">
 </applet>*/
public class Reverse extends Applet implements ActionListener
{
	Button b;
```

```
	TextField t,t1;
	public void init()
	{
		b=new Button("R");
		Label l=new Label("Enter a Number");
		add(l);
		t=new TextField(15);
		add(t);
		b.addActionListener(this);
		add(b);
	}
	public void actionPerformed(ActionEvent e)
	{
		String sa=t.getText();
		StringBuffer a = new StringBuffer(sa);
		showStatus(a.reverse().toString());
	}
}
```

Question 4:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="MouseEvents" height=300 width=300>
</applet>
*/
public class MouseEvents extends Applet{
	String msg="";
		public void init()
		{
			addMouseListener(new MouseAdapter(){
				public void mouseClicked(MouseEvent me){
				msg="WELCOME";repaint();
				}
				public void mouseReleased(MouseEvent me)
				{

				msg="GOODBYE";repaint();

				}
```

```
                        });
                }
                public void paint(Graphics g)
                {
                g.drawString(msg,10,10);
                }
}
```

Question 5:

```
import java.util.*;
class Name extends Exception
{       String name;
        Name(String n)
        {
                name=n;
        }
        public String toString()
        {       return name+" is not an acceptable name";
        }
}
class Age extends Exception
{
        int age;
        Age(int a)
        {
                age=a;
        }
        public String toString()
        {
                return age+" is greater than 50";
        }
}
class Employee
{       int a;String n;
        Employee(int a,String n)
        {
                this.a=a;
                this.n=n;
        }
```

```
	void display()
	{
		System.out.println("Employee Name:"+n+"\nEmployee age:"+a);
	}
}
public class Main
{
	public static void main(String args[])throws Age,Name
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter Name and age");
		int a,h;String n;
		boolean b=true;
		n=sc.next();
		a=sc.nextInt();
		try
		{
			h=Integer.parseInt(n);
			b=false;
			throw new Name(n);
		}
		catch(Name e)
		{
			System.out.println(e);
		}
		catch(Exception e)
		{
		}
		try
		{
			if(a>50)
			{	b=false;
				throw new Age(a);
			}
		}
		catch(Age e)
		{
			System.out.println(e);
		}
		if(b==true)
		{
			Employee e =new Employee(a,n);
			e.display();
```

```
                }
        }
}
```

Question 6:

```
class MathSin extends Thread {
    double deg,result;
    public MathSin(double degree)
    {
    deg=degree;
    }
    public void run()
    {
    result=Math.sin(deg);
    }


}
class MathCos extends Thread {

    double deg,result;
    public MathCos(double degree)
    {
    deg=degree;
    }
    public void run()
    {
    result=Math.cos(deg);
    }


}
class MathTan extends Thread {
    double deg,result;
    public MathTan(double degree)
    {
    deg=degree;
    }
    public void run()
    {
    result=Math.tan(deg);
    }
```

```
}
public class Main {

    public static void main(String[] args) {
        double totaladd;
        MathSin sin=new MathSin(45.0);
        MathCos cos=new MathCos(45.0);
        MathTan tan=new MathTan(45.0);
        sin.start();
        cos.start();
        tan.start();
        try{
            sin.join();
            cos.join();
            tan.join();
            totaladd=sin.result+cos.result+tan.result;

            System.out.println("sin(x) + cos(x) +tan(x) :"+totaladd);

        }catch(InterruptedException IntExp){

        }
    }

}
```

Question 7:

```
import java.util.*;
class Triangle extends Exception
{
        public String toString()
        {
                return "Not an acceptable triangle";
        }
}
public class Main
{
        public static void main(String args[])throws Triangle
        {       Scanner sc=new Scanner(System.in);
```

```
		int a,b,c;
		System.out.println("Enter the 3 sides of a triangle:");
		a=sc.nextInt();
		b=sc.nextInt();
		c=sc.nextInt();
		try
		{
			if((a<b+c)&&(b<a+c)&&(c<a+b))
				System.out.println("Valid Triangle");
			else
				throw new Triangle();
		}
		catch(Triangle e)
		{
			System.out.println(e);
		}
	}
}
```

Question 8:

```
import java.util.*;
public class Main
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		String s;
		System.out.println("Enter the word");
		s=sc.next();
		StringBuffer sb=new StringBuffer(s.toLowerCase());
		for(int i=0;i<sb.length();i++)
		{

			if(sb.charAt(i)=='a'||sb.charAt(i)=='e'||sb.charAt(i)=='i'||sb.charAt(i)=='o'||sb
			.charAt(i)=='u')
			{
				int x=s.charAt(i);
				x=x+1;
				sb.setCharAt(i,(char)x);
			}
		}
```

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

Question 9:

```
class A
{
        synchronized void displayTable(int n)
        {       try
                {
                        for(int i=1;i<=10;i++)
                        {
                                System.out.println(i*n);
                                Thread.sleep(1000);
                        }
                }
                catch(Exception e)
                {

                }
        }
}
class Mul extends Thread
{
        A a;int n;
        Mul(A a,int n)
        {       this.n=n;
                this.a=a;
        }
        public void run()
        {
                a.displayTable(n);
        }
}
public class Main
{
        public static void main(String args[])
        {
                A a=new A();
                Mul m=new Mul(a,8);
                Mul m1=new Mul(a,9);
```

```
                m.start();
                m1.start();
        }
}
```

Question 10:

```
interface Calc{
    public int res(int x,int y);
}

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

        Calc a=(x,y)->(x+y);
            System.out.println("\nSUM IS:"+a.res(2,3));


        Calc  m=(x,y)->(x*y);
            System.out.println("\nPRODUCT IS:"+m.res(3,4));
    }

}
```

Question 11:

```
import java.util.*;
public class Main
{
        public static void main(String args[])
        {
                Scanner sc=new Scanner(System.in);
                String s;
                System.out.println("Enter string");
                s=sc.nextLine();
                int c=0;
                char ch=s.charAt(0);
                for(int i=1;i<s.length();i++)
                {
                        if(s.charAt(i)==ch)
```

```
                    c++;
                ch=s.charAt(i);
            }
            System.out.println(c);
        }
}
```

Question 12:

```
public class Main extends Thread
{
            public static void main(String args[]) throws InterruptedException
            {
                Thread T1=new Thread();
                Thread T2=new Thread();
                Thread T3=new Thread();
                Thread T4=new Thread();
                Thread T5=new Thread();
                T1.setPriority(7);
                T2.setPriority(2);
                T3.setPriority(10);
                T4.setPriority(5);
                T5.setPriority(8);
                T1.start();
                if (T1.isAlive())
                    System.out.println("Thread 1 is alive");
                else
                    System.out.println("Thread 1 is not alive");
                T2.start();
                if (T2.isAlive())
                    System.out.println("Thread 2 is alive");
                else
                    System.out.println("Thread 2 is not alive");
                T3.start();
                T3.sleep(1000);
                if (T3.isAlive())
                    System.out.println("Thread 3 is alive");
                else
                    System.out.println("Thread 3 is not alive");
                T4.start();
                if (T4.isAlive())
                    System.out.println("Thread 4 is alive");
                else
                    System.out.println("Thread 4 is not alive");
```

```
                T5.start();
                T5.sleep(1500);
                if (T5.isAlive())
                        System.out.println("Thread 5 is alive");
                else
                        System.out.println("Thread 5 is not alive");
        }
}
```