Q1

class A {

public static void main(String args[]) throws InterruptedException {

WebCount obj = new WebCount();

//System.out.println(Thread.currentThread().getName());

Thread t1 = new Thread(new Runnable() {

//@Override

public void run() {

for (int i = 1; i <= 100; i++) {

obj.webcount();

}

}

});

Thread t2 = new Thread(new Runnable() {

//@Override

public void run() {

for (int i = 1; i <= 100; i++) {

obj.webcount();

}}

});

Thread t3 = new Thread(new Runnable() {

@Override

public void run() {

for (int i = 1; i <= 100; i++) {

obj.webcount();

}}

});

Thread t4 = new Thread(new Runnable() {

@Override

public void run() {

for (int i = 1; i <= 100; i++) {

obj.webcount();

}

}

});

Thread t5 = new Thread(new Runnable() {

@Override

public void run() {

for (int i = 1; i <= 100; i++) {

obj.webcount();

}

} });

t1.start();

t2.start();

t3.start();

t4.start();

t5.start();

t1.join();

t2.join();

t3.join();

t4.join();

t5.join();

System.out.println(WebCount.count);

}}

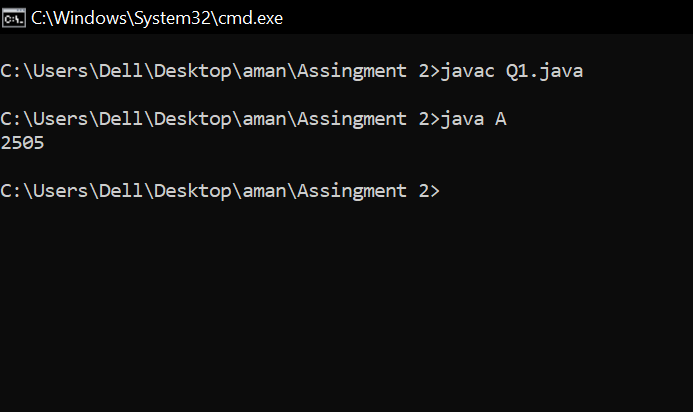
class WebCount {

public static int count = 5;

public synchronized void webcount() {

count+=5;

}

}

Q2

import javax.swing.\*;

import java.awt.\*;

class AB extends Canvas

{

public void paint(Graphics g)

{

g.drawOval(50,50,400,400);

g.setColor(Color.orange);

g.fillOval(50,50,400,400);

g.setColor(Color.black);

g.drawOval(140,140,50,50);

g.setColor(Color.black);

g.fillOval(140,140,50,50);

g.setColor(Color.black);

g.drawOval(300,140,50,50);

g.setColor(Color.black);

g.fillOval(300,140,50,50);

g.setColor(Color.black);

//g.drawLine(300,140,50,50);

g.drawArc(145,200,200,200,0,-180);

}

public static void main(String args[])

{

AB a=new AB();

JFrame f=new JFrame();

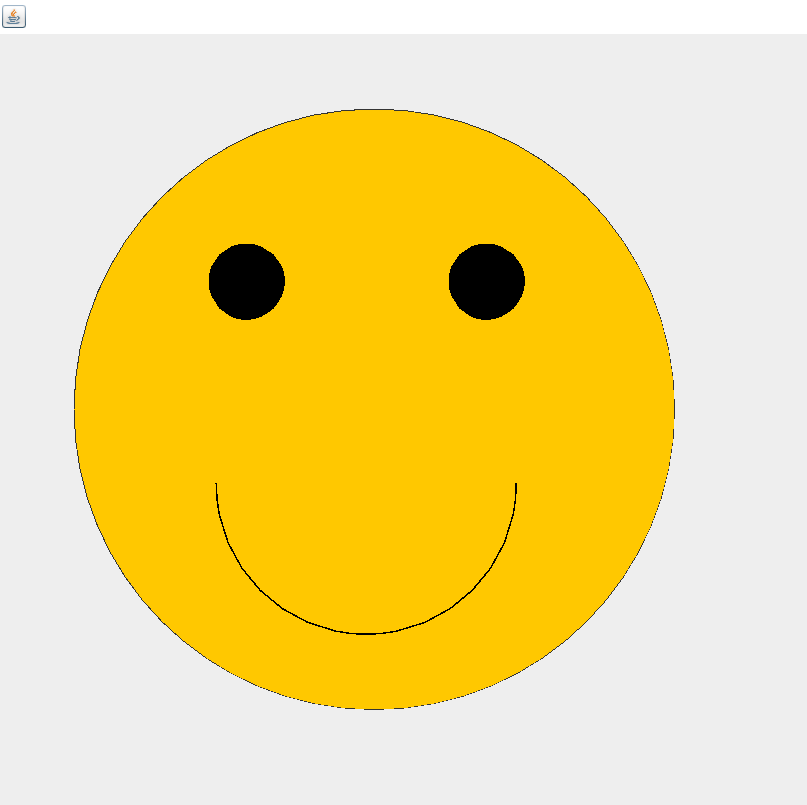
f.add(a);

f.setSize(400,400);

f.setVisible(true);

}

}



Q3

import javax.swing.\*;

import java.awt.\*;

class AB extends Canvas

{

public void paint(Graphics g)

{

g.setColor(Color.blue);

g.drawLine(0, 0, 400, 400);

g.setColor(Color.black);

g.drawOval(50,18,50,50);

g.setColor(Color.blue);

g.fillOval(50,18,50,50);

}

public static void main(String args[])

{

AB a=new AB();

JFrame f=new JFrame();

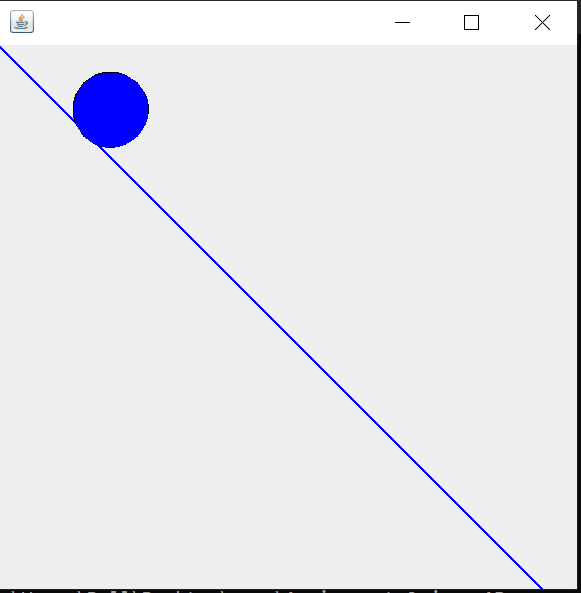
f.add(a);

f.setSize(400,400);

f.setVisible(true);

}

}



Q4

class A

{

int d,m,k,c,dd;

static int min=0,max=0;

static String s="",ct="";

A(int d,int m,int k,int c,int dd)

{

this.d=d;

this.m=m;

this.k=k;

this.c=c;

this.dd=dd;

if(d<=m&&d<=c&&d<=dd&&d<=k){

min=d;ct="Delhi";}

else if(m<=d&&m<=c&&m<=dd&&m<=k){

min=m;ct="Mumbai";}

else if(k<=d&&k<=m&&k<=dd&&k<=c){

min=k;ct="Kolkata";}

else if(c<=d&&c<=m&&c<=dd&&c<=k){

min=c;ct="Chennai";}

else{ min=dd;ct="Dehradun";}

if(d>=m&&d>=c&&d>=dd&&d>=k){

max=d;s="Delhi";}

else if(m>=d&&m>=c&&m>=dd&&m>=k){

max=m;s="Mumbai";}

else if(k>=d&&k>=m&&k>=dd&&k>=c){

max=k;s="Kolkata";}

else if(c>=d&&c>=m&&c>=dd&&c>=k){

max=c;s="Chennai";}

else{ max=dd;s="Dehradun";}

}

A()

{

System.out.println(min+" "+ct);

System.out.println(max+" "+s);

}

public static void main(String args[])

{

A d0=new A(24,4,15,35,60);

A d1=new A(14,10,2,5,6);

A d2=new A(4,5,3,2,30);

A d3=new A(4,15,35,2,40);

A d4=new A(26,1,35,35,60);

A d5=new A(24,15,2,5,60);

A d6=new A(24,15,3,35,60);

A d7=new A(0,15,35,4,60);

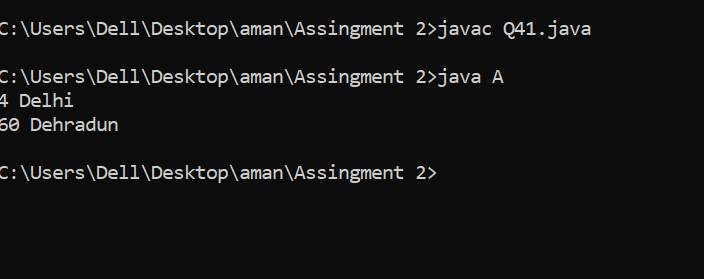
A d8=new A(24,10,3,35,60);

A d9=new A(4,15,8,15,60);

A obj=new A();

}

}



Q5

class A

{

int d,m,k,c,dd;

A(int d,int m,int k,int c,int dd)

{

this.d=d;

this.m=m;

this.k=k;

this.c=c;

this.dd=dd;

System.out.println((d+m+k+c+dd)/5);

}

public static void main(String args[]){

A d0=new A(24,4,15,35,60);

A d1=new A(14,10,2,5,6);

A d2=new A(4,5,3,2,30);

A d3=new A(4,15,35,2,40);

A d4=new A(26,1,35,35,60);

A d5=new A(24,15,2,5,60);

A d6=new A(24,15,3,35,60);

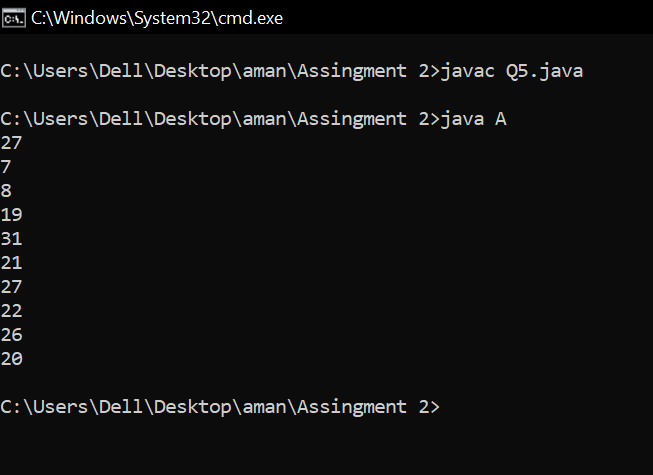
A d7=new A(0,15,35,4,60);

A d8=new A(24,10,3,35,60);

A d9=new A(4,15,8,15,60);

}

}



Q22

class PingPong {

public static void main(String[] args) {

Object LOCK\_OBJECT = new Object();

Thread ping = new Thread(new PingPongThread(LOCK\_OBJECT, "Ping"));

Thread pong = new Thread(new PingPongThread(LOCK\_OBJECT, "Pong"));

ping.start();

pong.start(); }}

class PingPongThread implements Runnable{

private Object LOCK\_OBJECT;

private String name;

public PingPongThread(Object LOCK\_OBJECT, String name) {

this.LOCK\_OBJECT = LOCK\_OBJECT;

this.name = name;}

@Override

public void run() {

synchronized (LOCK\_OBJECT) {

while(true) {

System.out.println(name);

try {

Thread.sleep(1000); } catch (InterruptedException e1) {

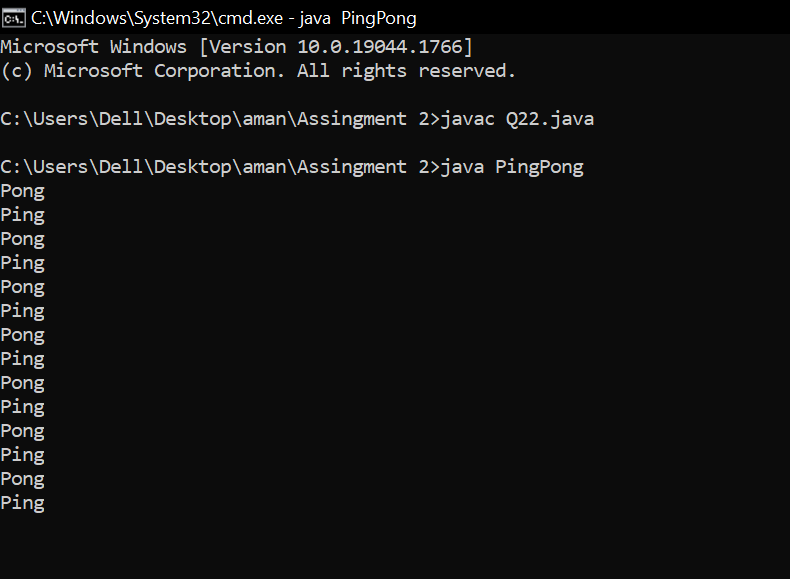
e1.printStackTrace(); }

LOCK\_OBJECT.notify(); try {

LOCK\_OBJECT.wait(1000);

} catch (InterruptedException e) {

e.printStackTrace();}}}}



Q23

class PingPong {

public static void main(String[] args) {

Object LOCK\_OBJECT = new Object();

Thread ping = new Thread(new PingPongThread(LOCK\_OBJECT, "Ping"));

Thread pong = new Thread(new PingPongThread(LOCK\_OBJECT, "Pong"));

ping.start();

pong.start();

}

}

class PingPongThread implements Runnable{

private Object LOCK\_OBJECT;

private String name;

public PingPongThread(Object LOCK\_OBJECT, String name) {

this.LOCK\_OBJECT = LOCK\_OBJECT;

this.name = name;

}

@Override

public void run() {

synchronized (LOCK\_OBJECT) {

while(true) {

System.out.println(name);

System.out.println(name);

try {

Thread.sleep(1000);

} catch (InterruptedException e1) {

e1.printStackTrace();

}

LOCK\_OBJECT.notify();

try {

LOCK\_OBJECT.wait(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

}

