

**Question 3**

a)

Improved performance due to locally available materials

Faster loading and better memory management

b)

i) True

ii) SQlite

Internal Storage

c)

i) Do not block the UI thread

ii) Do not access the Android UI toolkit from outside the UI thread

d)

i)

Primes

Find primes

Maximum:

ii)

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

inMax=findViewById(R.id.in\_max);

outPrimes=findViewById(R.id.out\_primes);

button=findViewById(R.id.button);

button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (!inMax.getText().toString().isEmpty()){

findPrimes();

}

}

});

}

iii)

private void findPrimes() {

int maxNo=Integer.parseInt(inMax.getText().toString());

if (maxNo==0){

Toast.makeText(this, "invalid maximum", Toast.LENGTH\_SHORT).show();

}else {

List<Integer> integers=calculatePrimes(maxNo);

for (Integer i:integers) {

System.out.println(i+",");

}

}

}

private static List<Integer> calculatePrimes(int max) {

boolean[] primeFlags = new boolean[max + 1];

Arrays.fill(primeFlags, true);

primeFlags[0] = primeFlags[1] = false;

for (int i = 2; i < primeFlags.length; i++) {

if (primeFlags[i]) {

for (int j = 2; i \* j < primeFlags.length; j++) {

primeFlags[i \* j] = false;

}

}

}

List<Integer> primes = new ArrayList<Integer>();

for (int i = 0; i < primeFlags.length; i++) {

if (primeFlags[i]) primes.add(i);

}

return primes;

}

**Question 5:**

a)

i)

import java.sql.\*;

public class Score {

public void main(String[] args) throws SQLException, ClassNotFoundException {

String url = "jdbc:derby://localhost/comps811u10";

String user = "APP";

String pass = "APP";

try (Connection conn = DriverManager.getConnection(url, user, pass);) {

Statement stmt = conn.createStatement();

String sql = "SELECT \* FROM Student ORDER BY Name";

ResultSet result = stmt.executeQuery(sql);

while (result.next()) {

int s = result.getInt("Score");

String n = result.getString("Name");

if (s >= 90) System.out.println(n);

}

}

}

}

ii)

Mary

Perter

b)

JPA allows you to use an object model in your application which can make your life much easier. JDBC allows you to do more things with the Database directly, but it requires more attention

c) Start of transcation

connection.setAutoCommit(false);

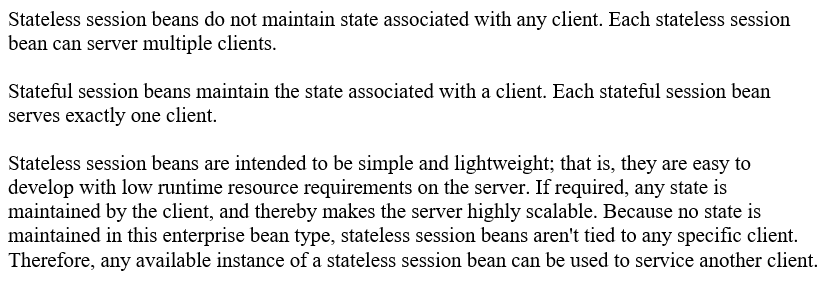
Rolling back incase of bas connection

connection.rollback();

On successful connection

connection.commit();

d)



e)

i)

package com.javatpoint;

import javax.ejb.Remote;

@Remote

public interface. TriangleImplRemote {

double area (double a, double b, double c);

}

ii)

package com.javatpoint;

import javax.ejb.Stateless;

@Stateless(mappedName = "Triangle")

public class TriangleImpl implements AdderImplRemote {

public double area (double a, double b, double c){

double s = (a + b + c) / 2;

return Math.sqrt(s \* (s-a) \* (s-b) \* (s-c));

}

}

iii)

package com.javatpoint;

import javax.naming.Context;

import javax.naming.InitialContext;

public class Test {

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

Context context=new InitialContext();

TriangleImplRemote remote=( TriangleImplRemote)context.lookup("Triangle ");

System.out.println(remote.add(3,4,5));

}

}