



eduresourceza.com

ST BENEDICT'S COLLEGE



SUBJECT
GRADE
EXAMINER
NAME
CLASS

Information Technology

11

Mrs Kench

DATE

MARKS

MODERATOR

DURATION

July 2014

120

3 Hours and 30min printing

COGNITIVE LEVELS

LOW ORDER

29 %

**MIDDLE
ORDER**

41 %

HIGH ORDER

30 %

INSTRUCTIONS
COMMENT

**TEACHER'S
SIGNATURE**
**PARENT'S
SIGNATURE**

QUESTION 1**SQL****40 MARKS**

A school is hosting a Shakespeare play festival. They have store the details of the actors, the plays and the roles in three tables.

- 1.1 List all the actors who earn more that R300 per hour in alphabetical order. (4)

```
SELECT tblActors.* ✓
FROM tblActors ✓
WHERE ActorRate >300 ✓
ORDER BY ActorName; ✓
```

- 1.2 The actors will all be paid R50 more per hour. Update the table tblActors to increase their rate of pay. (2)

```
UPDATE tblActors ✓
SET ActorRate = ActorRate + 50; ✓
```

- 1.3 Add a new Actor using your name, date of birth and any rate you choose. (3)

```
INSERT INTO tblActors(ActorName, ActorDOB, ActorRate) ✓
VALUES("Delia",#1990-01-01#,500) ✓ ✓
```

- 1.4 Create a query to determine the name and age of all the actors sorted according to age from oldest to youngest. Name the field as age. (6)

```
SELECT ActorName, ✓ year(now()) ✓ - year(ActorDOB) ✓ As age ✓
FROM tblActors
ORDER BY year(now()) - year(ActorDOB) ✓ DESC ✓
```

- 1.5 Count the number of roles in each show. (5)

```
SELECT ShowID, COUNT(*) ✓✓ as NumRoles
FROM tblRoles ✓
GROUP BY ShowID ✓✓
```

- 1.6 Calculate the total amount of money that should be brought in from all the shows considering the number of seats in each show and the capacity of a show for a single night. (4)

```
SELECT SUM ✓ (ShowCapacity*ShowCost) ✓✓ AS Total
FROM tblShows; ✓
```

- 1.7 Find the longest show on each day. Show only the shows that are longer than 200 minutes. (8)

```
SELECT tblShows.ShowDate ✓, Max ✓ (tblShows.ShowDuration) ✓ AS MaxOfShowDuration
FROM tblShows ✓
GROUP BY tblShows.ShowDate ✓✓
HAVING MAX(ShowDuration) > 200 ✓✓
```

- 1.8 List all the Actors who have a birthday in July and more than 13 characters in their combined name and surname. Show only the actor's name and date of birth. (5)

```
SELECT ActorName, ActorDOB ✓
FROM tblActors ✓
WHERE month(ActorDOB) = 7 ✓ AND ✓ LEN(ActorName) > 13; ✓
```

- 1.9 Find the average cost of each show. (3)

```
SELECT AVG(ShowCost) V As AVE
FROM tblShows
GROUP BY ShowNameVV
```

OOP

```
package memo;
```

```
import java.util.Scanner;
public class Term
{
```

Question 2.1 (4)

```
    private VString Vterm; V
    private String explan; V
```

Question 2.2 (4)

```
    public TermV (String term, String explan) V
    {
        this.term = term; V
        this.explan = explan; V
    }
```

Question 2.3 (2)

```
    public String getTerm()
    {
        return term; V
    }
```

```
    public String getExplan()
    {
        return explan; V
    }
```

Question 2.4 (4)

```
    @Override
    public String toString() V
    {
        return term V+ "\t" V + explan; V
    }
```

Question 2.5 (9)

```
    public Term V (String e) V
    {
        term = "";
        Scanner sc = new Scanner (e); V
        while (sc.hasNext()) V
        {
            term = termV + sc.next()V.charAt(0); V
        }
        term = term.toUpperCase(); V
        explan = e; V
    }

    public void setTerm(String term)
    {
        this.term = term;
    }
```

```

public void setExplan(String explan)
{
    this.explan = explan;
}

}

package memo;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
public class TermArray
{
Question 3.1 (5)
    private✓ Term tArr[]✓ = new Term[150];✓
    private✓ int size = 0;✓

Question 3.2 (9)
    public TermArray()
    {
        try
        {
            Scanner scFile = new Scanner(new File("Terms.txt"));✓
            String line, term, ex;
            while (scFile.hasNext())✓
            {
                line = scFile.nextLine();✓
                System.out.println(line);
                Scanner sc = new Scanner(line).useDelimiter("#");✓✓
                term = sc.next();✓
                ex = sc.next();✓
                System.out.println(term + "\t" + ex);
                sc.close();
                tArr[size] = new Term(term, ex);✓
                size++;✓
            }
            scFile.close();
        } catch (FileNotFoundException f)
        {
            System.out.println("File Not Found - check file name or path");

```

```

    }
}

```

Question 3.3 (5)

```

public String toString()✓
{
    String temp = "";
    for (int i = 0; i < size; i++)✓✓
    {
        temp = temp + i + " " + tArr[i] + "\n";✓
    }
    return temp;✓
}

```

Question 3.4 (8)

```

public void sort()
{
    for (int i = 0; i < size - 1; i++)✓✓
    {
        for (int j = i + 1; j < size; j++)✓✓
        {
            if (tArr[i].getTerm().compareTo(tArr[j].getTerm()) > 0) ✓✓
            {
                Term temp = tArr[i];✓✓
                tArr[i] = tArr[j];
                tArr[j] = temp;
            }
        }
    }
}

```

Question 3.5 (5)

```

public void insert✓ (String e) ✓
{
    tArr[size] = new✓ Term(e); ✓
    size++;✓
}

```

Question 3.6 (8)

```

public String search(String st) ✓
{
    int i = 0;✓
    while (i < size) ✓

```

```

{
    if (st.equalsIgnoreCase(tArr[i].getTerm()))✓
    {
        return tArr[i].getExplan();✓✓
    }
    i++;✓
    System.out.println("i " + i + " size " + size);
}
return null;✓
}

```

Question 3.7 (8)

```

public String decode(String s) ✓
{
    String word = "", temp = "", sen = "";
    Scanner sc = new Scanner (s); ✓
    while (sc.hasNext())✓
    {
        word = sc.next();✓
        System.out.println("The word is " + word);
        sen = search(word); ✓
        if (sen == null) ✓
        {
            temp = temp + word + " ";✓
        }
        else temp = temp + sen + " ";✓
    }
    return temp;
}
}

```

Question 4 (9)

```

public class TermUI
{
    public static void main(String[] args)
    {
        TermArray ta = new TermArray();✓✓
        System.out.println(ta); ✓
        String s = JOptionPane.showInputDialog("Enter explanation to add");✓
        ta.insert(s); ✓
        System.out.println(ta);

```

```
ta.sort();✓  
System.out.println(ta);✓  
String sen = JOptionPane.showInputDialog("Input sentence to  
decode");✓  
System.out.println(ta.decode(sen));✓  
}  
}
```
