



eduresourceza.com

ST BENEDICT'S COLLEGE



SUBJECT	Information Technology Theory	DATE	July 2015
GRADE	12	MARKS	120
EXAMINER NAME	Mrs Kench	MODERATOR	Mr Gill
CLASS		DURATION	3 Hours and 30min printing

COGNITIVE LEVELS			
LOW ORDER	30 %	MIDDLE ORDER	40 %
		HIGH ORDER	30 %

INSTRUCTIONS COMMENT	
TEACHER'S SIGNATURE	
PARENT'S SIGNATURE	

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

1.1:

```
SELECT CountryName, Population ✓
FROM Country ✓
ORDER BY ✓ Population DESC ✓, CountryName
```

(4)

1.2:

```
SELECT CityName
FROM City
WHERE (Population > 100000) ✓ AND (Capital = TRUE) ✓ AND (Latitude BETWEEN -20 AND 20) ✓
ORDER BY CityName ✓
```

(4)

1.3:

```
INSERT INTO City ( CityName, Province, Capital, Population ) ✓
SELECT 'Dudley Heights', Province, Capital, Population
FROM City ✓✓
WHERE CityName = 'Colchester' ✓
```

(5)

1.4:

```
UPDATE ✓ City SET Latitude = 0.1, Longitude = 0.1 ✓
WHERE (Latitude IS NULL ✓ ) AND (Longitude IS NULL) ✓
```

(4)

1.5:

```
SELECT CountryName, YEAR(NOW()) ✓ - YEAR(Census) ✓ AS [Years Since Census] ✓
FROM Country
```

(3)

1.6:

```
SELECT CountryName, COUNT(*) ✓ AS [Number of Cities]
FROM Country
INNER JOIN ✓ (Province INNER JOIN City ON Province.ID = City.Province) ✓
ON Country.Code = Province.Country ✓
GROUP BY ✓ CountryName HAVING ✓ COUNT(*) > 1 ✓
```

(7)

1.7:

```
SELECT CountryName, Country.Area, SUM(Province.Area) ✓,
ROUND ✓ ((Country.Area-SUM(Province.Area))/Country.Area*100,2) ✓✓ AS Variance
FROM Country
INNER JOIN ✓ Province
ON Country.Code = Province.Country ✓
GROUP BY ✓ CountryName, Country.Area ✓
```

1.8:

```
SELECT TOP 3 ✓ ProvinceName, Population
FROM Province
WHERE Population < ✓ (SELECT AVG(Population) FROM Province) ✓✓
ORDER BY Population DESC✓
```

(5)

Total: 40**SECTION B****OOP****80 MARKS****QUESTION 2****28 MARKS**

```
public class Numbers {
```

2.1 (3)

```
    private✓ int[] numArr✓ = new int[6]; ✓
```

2.2 (4)**2.5 (6) can be a separate method**

```
    public Numbers()✓{
        String temp = "#";
        int num = 0;
        for (int i = 0; i < 6; i++)✓ {
            do {✓
                num = (int) (Math.random() * 50 + 1); ✓✓
            } while (temp.indexOf✓ (num + "#") > 0); ✓
            temp = temp + num + "#"; ✓✓
            numArr[i] = num; ✓
        }
    }
}
```

2.6 (5)

```
    public int ✓countNumbers(String numStr) ✓ {
        int count = 0;
        for (int i = 0; i < 6; i++)✓ {
            if (numStr.indexOf(" " + numArr[i]) > 0) {✓
                count++;✓
            }
        }
        return count;
    }
}
```

2.3 (4)

```
    public String toString() {✓
        String temp = "";
```

```

for (int i = 0; i < 6; i++) {✓
    temp = temp + numArr[i] + " ";✓
}
return temp; ✓
}

```

2.4 (6)

```

public void sort() {
    for (int i = 0; i < 5; i++) {✓
        for (int j = i + 1; j < 6; j++) {✓
            if (numArr[i] > (numArr[j])) {✓✓
                int temp = numArr[i];
                numArr[i] = numArr[j];
                numArr[j] = temp; ✓✓
            }
        }
    }
}

```

QUESTION 3**19 MARKS**

```

public class Ticket {

```

3.1 (4)

```

    private✓ String date, time; ✓
    private Numbers nArr; ✓✓

```

3.2 (4)

```

    public Ticket(String d, String t) ✓
    {
        date = d;
        time = t; ✓
        nArr = new Numbers(); ✓✓
    }

```

3.4 (5)

```

    public String toString()
    {
        nArr.sort(); ✓
        return✓ ">>>" + date + ">>>"✓ + changeTime()✓ + "****" +
        nArr.toString() + "\n"; ✓
    }

```

3.3 (6)

```

    public String changeTime()
    {

```

```

String temp = time;
String hour = temp.substring(0,2);
String min = temp.substring(3); ✓
int intHour = Integer.parseInt(hour); ✓
if(intHour > 12) ✓
{
    intHour = intHour - 12; ✓
    temp = intHour + ":" + min + "pm"; ✓
}
else temp = temp + "am"; ✓
return temp;
}

```

6.1 (3)

```

public int count(String st) ✓
{
    return nArr.countNumbers(st); ✓✓
}
}

```

QUESTION 4**15 MARKS****4.1 (3)**

```

public class TicketArray {
    private ✓ Ticket tArr[] = new Ticket[200]; ✓
    private int size = 0; ✓
}

```

4.2 (8)

```

public TicketArray()
{
    try
    {
        Scanner scFile = new Scanner(new
File("LottoTickets.txt")); ✓
        String line, d,t;
        while (scFile.hasNext()) ✓
        {
            line = scFile.nextLine(); ✓
            System.out.println(line);
            Scanner sc = new Scanner(line).useDelimiter("#"); ✓
            d = sc.next();
            t = sc.next(); ✓

            System.out.println(d + "\t" + t );
            sc.close();
            tArr[size] = new Ticket(d,t); ✓✓
            size++; ✓
        }
    }
}

```

```

        scFile.close();
    } catch (FileNotFoundException f)
    {
        System.out.println("File Not Found - check file name or
path");
    }
}

```

6.1 (10)

```

public String checkTickets(String st)
{
    String temp= "";
    Ticket hiTicket = tArr[0]; ✓
    int hi = tArr[0].count(st); ✓
    for (int i = 0; i < size; i++) ✓
    {
        if(tArr[i].count(st) > hi) ✓
        {
            hiTicket = tArr[i]; ✓
            hi = tArr[i].count(st); ✓
        }
    }
    for (int i = 0; i < size; i++) { ✓
        if(tArr[i].count(st) == hi) ✓
            temp = temp + tArr[i] + "\n"; ✓
    }
    return temp; ✓
}

```

4.3 (4)

```

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

```

QUESTION 5**3 MARKS**

```

public class LottoProj {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

```

5.1 (2)

```
TicketArray ta = new TicketArray(); ✓✓
```

5.2 (1)

```
System.out.println(ta); ✓
```

6.1 (2)

```
String str = JOptionPane.showInputDialog("Enter the 7  
numbers"); ✓  
System.out.println("The winning numbers are " + str);  
System.out.println("The tickets with the most numbers are: " +  
ta.checkTickets(str)); ✓  
}  
  
}
```

eduresourceza.com