



Learner's Name: _____

INFORMATION TECHNOLOGY

PAPER 1 – PRACTICAL

JUNE EXAMINATION – 2013

MEMORANDUM AND MARKING GRID

Q1	Q2	Q3	Total
[38]	[55]	[27]	[120]

Question 1: Java Programming and Database

1.1	<u>Option A: Display All Members</u> SELECT * ✓ FROM Members ✓ ORDER BY ✓ MemSName, MemFName (Correct order)✓	4
1.2	<u>Option B: Members with Birthdays in June</u> SELECT MemFName, MemSName, MemCellNr, MemBDate FROM Members✓ (YEAR(NOW())- ✓ (or 2013 -) YEAR(MemBDate))✓ AS Age ✓ WHERE MONTH(MemBDate)✓ = 6 ✓ ORDER BY DAY(MemBDate)✓	7
1.3	<u>Option C: Members not on Gmail</u> SELECT MemEmail, MemFName, MemSName, MemCellNr FROM Members ✓ WHERE MemEmail ✓ NOT✓ LIKE✓ '%gmail%'✓ (for % %)	5
1.4	<u>Option D: Number of Members Not Paying Fees</u> SELECT COUNT(*)✓ AS NumMembersWithFeesOutstanding✓ FROM Members WHERE MemTypeCode = 'SGM1' ✓ AND ✓ MemFeeOutstanding > 0✓	5
1.5	<u>Option E: Delete a Member from the Database</u> DELETE FROM Members ✓ WHERE MemID = ✓ '" + input + "'✓✓	4
1.6	<u>Option F: View Members Profile</u> SELECT MemID, MemFName, MemSName, MemTMFee ✓ FROM Members m✓, MShipTypes t✓ (aliases m and t optional) WHERE m.MemTypeCode = t.MemTypeID ✓ AND m.MemID = '" + input + "'✓✓ (or FROM Members m INNER JOIN MShipTypes t ON m.MemTypeCode = t.MemTypeID WHERE m.MemID = '" + input + "'✓)	6
1.7	<u>Option G: Insert New Member</u> INSERT ✓ INTO Members ✓ (MemID, MemBDate, MemFName, MemSName, MemID, MemEmail, MemTypeCode, MemFeesOutstanding, MemJoinDate) - optional as <u>all</u> fields are being used, NO mark VALUES ✓ ('GST20185', #31/05/1953#, 'Sibusiso', 'Thembani', '0768562655', 'SThembani@gmail.com', 'SGM3', 30.0, #2013/05/30#) All strings (incl. ID) in ' '✓, both dates in # #✓, double 30.0 none✓, correct order✓	7
Total Question 1:		[38]

Question 2: Java – Object-oriented Programming			
2.1.1	<pre>public class GymMember { private String fName; private String sName; private String idNum; private double feesOwing; }</pre>	Class declaration✓ Suitable types✓ (-1 if NOT private as stated in the paper)	2
2.1.2	<pre>public GymMember(String fName, String sName, String idNum, double feesOwing) { this.fName = fName; this.sName = sName; this.idNum = idNum; this.feesOwing = feesOwing; }</pre>	4 parameters of correct types✓ 4 correct assignments✓	2
2.1.3	<pre>public String getFName() { return fName; } public String getSName() { return sName; } public double getFeesOwing() { return feesOwing; } //public String getIdNum() { return idNum; } OPTIONAL (not asked)</pre>	1 mark each, correct types and returns (-1 if not 100% correct)✓✓✓	3
	<pre>public void setFName(String fName) { this.fName = fName; } public void setsName(String sName) { this.sName = sName; } public void setIdNum(String idNum) { this.idNum = idNum; } public void setFeesOwing(double feesOwing) { this.feesOwing = feesOwing; }</pre>	OPTIONAL , not asked, don't penalise if done	
2.1.4	<pre>public char getGender() { int genderCode = Integer.parseInt(idNum.substring(6, 7)); if(genderCode < 5) return 'f'; else return 'm'; }</pre>	Of type char✓ Substring (6,7)✓ parseInt✓ if✓ return 'f' or 'm'✓	5
2.1.5	<pre>public String toString() { return "First Name = " + fName + "\nSurname = " + sName + "\nID Num = " + idNum + "\nGender = " + getGender() + "\nFees Owing = " + String.format("%s%.2f", "R", feesOwing) + "\n"; }</pre>	Labels and variables✓ Call of getGender()✓ New line✓ R in front of fees✓ Fees rounded to 2 decimals✓ Returns a string✓	6
Subtotal 2.1 =			[18]

2.2	In GymApp.java class:		
2.2.1	GymMember[] memberArr = new GymMember[1000];	Declaration of array of correct type✓ for 1000 elements✓	2
2.2.2	<pre> int count = 0; File txf = new File("MemberData.txt"); try { Scanner sc = new Scanner(txf); OR BufferedReader br = new BufferedReader(new FileReader(txf)); while (sc.hasNext()) { OR while (line = br.readLine() != null) OR while (br.ready()) { String line = sc.nextLine(); OR String line = br.readLine(); String[] parts = line.split("#"); OR Scanner sc2 = new Scanner(line).useDelimiter("#"); String fName = parts[0]; OR sc2.next(); String sName = parts[1]; OR sc2.next(); String id = parts[2]; OR sc2.next(); double fees = Double.parseDouble(parts[3]); OR sc2.nextDouble(); GymMember m = new GymMember(fName, sName, id, fees); memberArr[count] = m; count++; } } System.out.println("Added to array: " + count + " members"); } catch (FileNotFoundException e) { System.out.println("File not found"); System.exit(0); } catch (Exception e) { e.printStackTrace(); } </pre>	Counter variable✓ File declaration Try..catch (or if (file.exists()))OPTIONAL, not asked A file reading object✓✓ (Scanner or BufferedReader) While loop✓ Correct condition✓ Read line from file✓ Split line on "#"✓ Assign 3 string variables✓ Assign 1 double variable✓ Instantiate new object✓ with 4 parameters✓ (can use set methods if coded in class) Increment counter✓ Assign object to array✓ Feedback OPTIONAL If not found Message OPTIONAL Exit not asked	13
(a)	case 'A': <pre> System.out.println("MEMBERS\n====="); for (int i = 0; i < count; i++) { System.out.println(memberArr[i]); } </pre>	Heading✓ For-loop✓ Print obj✓ (toString() OPTIONAL)	3
(b)	case 'B': <pre> System.out.println("\nAlphabetical selection:\n====="); System.out.print("Enter a letter of the alphabet: "); char letter = keyb.next().toUpperCase().charAt(0); for (int i = 0; i < count; i++) { if (memberArr[i].getName().startsWith(""+letter)) { System.out.println(memberArr[i].getName() + ", " + memberArr[i].getfName()); } } </pre>	Heading OPTIONAL Input✓ char✓ in one case✓ (either upper or lower) For-loop✓ Use get SurName✓ First letter✓ (startsWith or substring(0,1)) = input✓ Print SurName, FirstName using gets✓	8
(c)	case 'C': <pre> int countNot0 = 0; double total = 0; for (int i = 0; i < count; i++) { if (memberArr[i].getFeesOwing() > 0) { countNot0++; total += memberArr[i].getFeesOwing(); } } double ave = total / countNot0; System.out.println(String.format("%d%sR%-3.2f%sR%-3.2f%s", countNot0, " members owe a total of ", total, " at an average of ", ave, " each.")); </pre>	Counter variable✓ Total variable✓ For-loop✓ Check feesOwing > 0✓ using get method✓ Inside IF: Increment counter✓ Add feesOwing to total✓ Work out average✓ (make sure one of the variables is double✓) Doubles to 2 decimals✓ Output✓	11
Subtotal 2.2 =			[37]
Total Question 2:			[55]

Question 3: Problem-Solving			
3.	<pre>import java.util.Scanner; import java.io.*; import javax.swing.*; // for JOptionPane public class RunningTraining { // Implementing a solution using two arrays, one for // months, one for distances, so-called "parallel arrays" private String[] months = {"July", "August", "September", "October", "November", "December", "January", "February", "March", "April", "May", "June"}; private int[] distances = {90, 100, 103, 145, 77, 135, 90, 107, 99, 128, 23, 177};</pre>	All given	
3.1	<pre>public void displayLog System.out.println ("Month Distance"); for (int i = 0; i < months.length; i++) { System.out.printf ("%12s%d\n", months[i], distances[i]); // OR println with "\t" } }</pre>	Headings✓ Loop through months or distances or 12✓ Output✓ with a clear space between months and distances✓	4
3.2	<pre>public void editLog(){ // Interface to update a month - 2 inputs: month's name, // updates km Scanner keyb = new Scanner (System.in); // OR BufferedReader System.out.print ("Please enter the month you want to update: "); // JOptionPane could also be used String inMonth = keyb.nextLine(); System.out.print ("Please enter the new number of kilometers: "); // JOptionPane could also be used int inDistance = keyb.nextInt(); boolean found = false; // similar flag can be used instead of boolean, e.g. an int for (int i = 0; i < months.length; i++) { if (inMonth.equalsIgnoreCase(months[i])) { found = true; distances[i] = inDistance; System.out.println (months[i]+" updated"); } } if (!found) System.out.println ("Month entered does not exist or was spelt incorrectly"); }</pre>	Prompting and reading month✓ Prompting and reading distance into int variable✓ Declare flag variable (boolean or int or even String)✓ Loop✓ Inside loop: Check input = months✓ using ignoreCase✓ If same month: Set flag to found✓ Set distance at same index✓ to new distance✓ Feedback✓ Check flag/if not found✓ Message for wrong month✓	12
3.3	<pre>public void logReport(){ int total = 0; for (int i = 0; i < distances.length; i++) { total += distances[i]; String out = " "; if (distances[i] >= 120) out = "achieved"; System.out.printf ("%12s%3s\n", months[i], out); } double average = total / 12.0; System.out.printf ("%s%.2f%s\n", "Over the last 12 months Jim ran an average of ", average, " km per month.\n"); }</pre>	Declare and initialise 'total'✓ For-loop✓ In loop: Add distance[i] to total✓ Test if distance[i] >= 120✓ Output month for all months✓ Output "achieved" if applicable✓ Calculate average✓ as double✓ Output average in sentence✓ Average 2 decimals OPTIONAL	11
Total Question 3:			[27]