

Information Technology P1 Practical Exam June 2015 Marking grid		Learner Name: _____	Q1	Q2	Q3	Total	Comment:	
			48	59	43	150		
							Max Mark	Mark Obtained
1.1	lblBanner.setBorder(null); lblBanner.setOpaque(true); lblBanner.setBackground(Color.black); lblBanner.setForeground(Color.white); lblBanner.setFont(lblBanner.getFont().deriveFont(24.0f)); Property sheet and/or code in constructor	Border removed ✓ Change label background colour to black ✓ Change labels transparency to Opaque = true ✓ Text colour white. ✓ Font size 24 ✓					1 1 1 1 1 5	
1.2.1	In rbnFirstNameActionPerformed txfFirstName.setEnabled(true); txfSurname.setEnabled(false); In rbtnSurnameActionPerformed txfFirstName.setEnabled(false); txfSurname.setEnabled(true); In rbnBothActionPerformed txfFirstName.setEnabled(true); txfSurname.setEnabled(true);	When Firstname Only is selected: Firstname text field enabled✓ Surname textfield disabled✓ When Surname Only is selected: Firstname text field disabledeach set in correct event ✓handler✓ Surname textfield enabled } When Both... is selected Both text fields enabled					1 1 1 1 4	
1.2.2	String firstname = txfFirstName.getText(); String surname = txfSurname.getText(); String fullname = firstname + surname; fullname = fullname.toUpperCase(); String letters = ""; for (int i = 0; i < fullname.length(); i++) { char letter = fullname.charAt(i); if (!(letter == 'A' letter == 'E' letter == 'I' letter == 'O' letter == 'U')) { letters += letter; } } txfNoVowels.setText(letters); Alternative solution to code in bold fullname = fullname.replace("A", ""); fullname = fullname.replace("E", ""); fullname = fullname.replace("I", ""); fullname = fullname.replace("O", ""); fullname = fullname.replace("U", "");	When remove vowels button clicked: Text from captured from text fields ✓ Concatenate surname to firstname ✓ Text converted to uppercase ✓ Remove vowels from text Loop ✓ Test condition ✓ Concatenation ✓ + ✓ Or any other algorithm to remove vowels (see left, 3 marks) Answer displayed in text Field (Not clear in question)					1 1 1 4 7	

1.3.1	<pre>String plateChars = txfNoVowels.getText(); can be global int numChars = (int) spnChars.getValue(); int numDigits = (int) spnDigits.getValue(); if (numChars > plateChars.length()) { javax.swing.JOptionPane.showMessageDialog(null, "The number of letters should be <= " + plateChars.length()); return; } if ((numDigits + numChars) > 7) { javax.swing.JOptionPane.showMessageDialog(null, "The total number of characters should be <= 7"); return; } btnGeneratePlate.setEnabled(true);</pre>	<p>Read and store No-Vowel string✓ [Give the mark if String in global variable] Read✓ and store ✓the spinner values</p> <p>If✓condition✓ Message dialog with error message✓ Validate if sum of letters and digits <= 7 If✓condition✓ Message dialog with error message✓ [One compound test using logical or () only 5 marks as no specific error message can be given]. Method exists correctly on error (no need for second test if first test fails)✓ Correct if else structure OR return used as in code example (do not accept System.exit(0) for return here) Enable the Generate number plate button when both tests pass✓</p>	1 2 2 1 2 1 1 1	10	
1.3.2	<pre>int numChars = (int) spnChars.getValue(); int numDigits = (int) spnDigits.getValue(); String plateChars = txfNoVowels.getText(); String prefix = plateChars.substring(0, numChars); String digits = ""; for (int i = 0; i < numDigits; i++) { digits += (int) (Math.random() * 10); } String numberPlateStr = prefix + " " + digits + " WP"; txfNumberPlate.setText(numberPlateStr);</pre> <p>Alternative random number generation: int digits = (int)(Math.random()*Math.pow(10,numDigits)); int zeros = numDigits - (""+digits).length(); String zeroString = ""; for(int i = 0; i < zeros; i++) { zeroString = zeroString + "0";} digits = zeroString + digits;</p>	<p>Read the required data again: number of letter, digits and the no vowel string✓ [Give the mark if these are available as global variables]</p> <p>Isolate the correct number of letters from noVowel string✓ Generate the correct number of random digits Digits randomly generated ✓✓ Allows for leading zeros ✓✓</p> <p>Number plate string concatenated correctly, spaces between letters, digits and WP✓ Number plate string displayed in txfNumberPlate✓</p> <p>[Also accept any algorithm that generate the correct number of digits which allows for leading zeros 4 marks]</p>	1 1 2 2 1 1 8		

1.4.1	<pre>int numChars = (int) spnChars.getValue(); int numDigits = (int) spnDigits.getValue(); int billableChars = numChars + numDigits; int registrationFee = 69; double cost = registrationFee; switch (billableChars) { case 1: cost += 10000.00; break; case 2: cost += 6000.00; break; case 3: cost += 4000.00; break; default: cost += 1750.00; } String costStr = String.format("%s R%.2f", lblCost.getText(), cost); lblCost.setText(costStr);(any formatting method)</pre>	<p>Billable character length: number of letters+number of letters excluding WP✓ Registration fee added✓</p> <p>Correct price assign ✓based on billable character length✓</p> <p>Text read from lblcost [or hard coded]. ✓Correctly formatted cost ✓appended. Displayed in label✓(Accept currency without ".00")</p>	<p>1</p> <p>1</p> <p>2</p> <p>2</p> <p>1</p>	7	
1.4.2	<pre>bngNames.clearSelection(); txfFirstName.setEnabled(false); txfSurname.setEnabled(false); btnGeneratePlate.setEnabled(false); txfNumberPlate.setText("ZZZ 9999 WP"); lblCost.setText("Cost of number plate:"); txfNoVowels.setText(""); spnChars.setValue(0); spnDigits.setValue(0);</pre>	<p>All radio buttons deselected✓ Name and surname fields disabled✓</p> <p>Generate Number Plate button disabled✓ Text in number plate text field set to ZZZ 9999 WP✓ [also accept empty string ""] Label cost text set to Cost of number plate:✓ Text field no vowels cleared✓ Spinners set to zero✓</p>	<p>1</p> <p>1</p> <p>1</p> <p>1</p> <p>1</p> <p>1</p> <p>1</p>	7	
Question 1 Total				48	
2.1.1	<pre>private String name; private int usedCarsSold; private int newCarsSold; private String makeLog; private String make;</pre>	<p>The keyword private ✓ Correct data types ✓ [Names were given] (ignore any additional attributes; regard as global variable)</p>	<p>1</p> <p>1</p> <p>2</p>		
2.1.2	<pre>public Salesperson(String name, String make) { this.name = name; this.make = make; this.makeLog = ""; }</pre>	<p>Constructor header with correct parameters ✓ Initialised relevant attributes using the parameters ✓ Initialised makeLog string to empty string ✓ (Do not penalise if remaining attributes were also given default values)</p>	<p>1</p> <p>1</p> <p>1</p> <p>3</p>		
2.1.3	<pre>public int getNewCarsSold() { return newCarsSold; }</pre>	<p>Accessor methods for correct attributes name and newCarsSold✓ Correct return type ✓</p>	<p>1</p> <p>1</p> <p>2</p>		
2.1.4	<pre>public void incrementUsedCarTotal() { usedCarsSold++; } public void incrementNewCarTotal() { newCarsSold++; }</pre>	<p>Methods uncommented✓</p>	<p>1</p> <p>1</p>		

2.1.5	<pre>public void updateLog(String make) { makeLog = makeLog + "#" + make; }</pre>	Parameter of method concatenated to makeLog attribute✓#delimiter inserted✓ (DO not accept makeLog = "#" + makeLog + make, (the make must be separated by "#". The string can start with a "#")	2	2	
2.1.6	<pre>public int carsOfMake() { Scanner textScanner = new Scanner(makeLog).useDelimiter("#"); int count = 0; while (textScanner.hasNext()) { if (textScanner.next().equals(make)) { count++; } } return count; }</pre>	Set up string scanner /split into array✓"#"✓ Loop through string or array ✓ Test ✓and Tally the number of occurrences of make in the string or array✓ Return this number✓ <i>Any algorithm that returns the correct number</i>	2 1 1 1 1	6	
2.1.7	<pre>public String toString() { return String.format("%-11s%-11s%-11s%-11s%-11s\n", name, (newCarsSold + usedCarsSold), newCarsSold, usedCarsSold, carsOfMake()); }</pre>	toString method header✓ and return✓ String formatted in neat columns✓ Sum to obtain total car sales per person✓ Call the carsOfMake✓ Includes only the required attributes✓ do not give mark if makelog and/or make is included (Do not penalise if %n or \n missing) (Accept tabs for it worked for the data in the paper) Examiner must ensure that tabs don't work to force formatting as given in the solution	2 1 1 1 1	6	
2.2.1	<pre>int[] newCarSales = new int[salesperson.length];</pre> <p>values should have been number in the question</p>	Array instantiated✓(accept int and double; Also accept new int[10]) Array length equal to the length of the salesperson array✓ If hardcoded using values in question 2.2.3 give length mark Array accessible to both buttons on the form✓ (if declared and created locally; in each method : Do not give mark) (If declared globally but created locally give the mark)	1 1 1	3	
2.2.2	<pre>File file = new File("CarSales.txt");(anonymous object in Scanner instantiation) (not required in BufferedReader) String makeOfCar = (String)cmbMake.getSelectedItemAt(); txaSalesTotals.setText(String.format("%-11s%-11s%-11s%-11s%-11s\n\n", "Name", "Cars Sold", "New cars", "Used Cars", makeOfCar)); int index = 0 try { for (String person : salesperson) { Salesperson salesPerson = new Salesperson(person, makeOfCar); Scanner fileScanner = new Scanner(new FileReader(file)); while (fileScanner.hasNext()) { String line = fileScanner.nextLine(); } } }</pre>	Create file object Read ✓and store make of car as selected in combo box (see creation of salesPerson object) Insert headings neat columns✓ and clearing previous text✓ index for parallel array✓(for int i = 0 ; ... also provides the index) Loop through salesperson array✓ Create object ✓ Open file for reading : reader/scanner and passing the filename/File object✓✓ Loop through file/ test for end of file✓ Read a line ✓	1 2 1 1 1 2 1 1		

	<pre> Scanner textScanner = new Scanner(line).useDelimiter(","); String name = textScanner.next(); String make = textScanner.next(); int year = textScanner.nextInt(); //or String year = textScanner.next() if (name.equals(salesPerson.getName())) { if (year == 2015) { salesPerson.incrementNewCarTotal(); } else { salesPerson.incrementUsedCarTotal(); } salesPerson.updateLog(make); } txaSalesTotals.append(salesPerson.toString()); newCarSales[index] = salesPerson.getNewCarsSold(); index++; fileScanner.close(); } } catch (FileNotFoundException ex) { JOptionPane.showMessageDialog(rootPane, "File not found"); System.exit(0); } </pre>	<p>Split/ instantiate Scanner ✓ delimiter “,” ✓ Read token strings ✓</p> <p>Read int/ parseInt if split used ✓ (can be String) Test if name equal salesperson ✓ Test if the car is new ✓ inside true block ✓ (if String: use .equals) Call objects increment methods to Increment used cars or new cars ✓ Call updateLog to update the log string ✓ Call the toString method after while loop. Display stats in text area ✓ Update new cars sold in parallel array ✓ Increment index for parallel array ✓ (also inside for i...) Close file ✓</p> <p>Catch: Message if file not found ✓ Exit program ✓</p>	2 1 1 1 2 1 1 1 1 1 1 1 1	25	
2.2.3	<pre> for (int i = 0; i < salesperson.length; i++) { for (int j = 0; j < salesperson.length - 1; j++) { int tempNumber; String tempName; if(newCarSales[j] < newCarSales[j+1]){ tempNumber = newCarSales[j]; newCarSales[j] = newCarSales[j+1]; newCarSales[j+1] = tempNumber; tempName = salesperson[j]; salesperson[j] = salesperson[j+1]; salesperson[j+1] = tempName; } } } for (int i = 0; i < 4; i++) { txaNewCarSalesTeam.append(salesperson[i] + "\n"); } </pre>	<p>Nested loop ✓ ✓</p> <p>Test totals ✓ Swap if required ✓ ✓ (Accept both descending and ascending order)</p> <p>Swap the salesperson array when totals are swapped ✓</p> <p>Read and display four names that sold the most new cars in text area ✓ ✓ taken from the bottom or top depending how it was sorted</p>	2 1 2 1 2	8	
2.2.4	<pre> btnSalesActionPerformed(null); </pre>	<p>Call the Sales Totals per Person event handler ✓ Accept or (evt)</p>	1	1	
Question 2 Total				59	

3.1	<pre>int total = 0; for (String vin : vinArr) { char year = vin.charAt(9); if (year == 'D' year == 'E') { total++; } } txfNumberOfCars.setText("" + total);</pre>	<p>Declare and Initialise counter to 0✓ Loop through vin number array✓ Identify the tenth character✓ [Test character against 'D' and 'E' (2013 and 2014)✓ ✓ Increment counter✓ Display final count in text field✓</p>	1 1 1 2 1 1	7	
3.2	<pre>private int getNumericalValue(char vinChar) { int value; switch (vinChar) { case 'A': case 'J': value = 1; break; case 'B': case 'K': case 'S': value = 2; break; case 'C': case 'L': case 'T': value = 3; break; case 'D': case 'M': case 'U': value = 4; break; case 'E': case 'N': case 'V': value = 5; break; case 'F': case 'W': value = 6; break; case 'G': case 'P': case 'X': value = 7; break; case 'H': case 'Y': value = 8; break; case 'R': case 'Z': value = 9; break; default: value = Integer.parseInt("" + vinChar); } return value; }</pre>	<p>Method to convert vin character to numerical value</p> <ul style="list-style-type: none"> • Parameter for the vin character ✓return type int✓ • Setup structure for numerical values if-else or switch case✓ • Test the vin character received✓ • Assign numerical value✓ • Convert digits character to correct value✓ • Return the numerical value✓ <p>(Also accept if the whole VIN was sent as a parameter and a corresponding string of numerical values was returned)</p>	2 1 1 1 1 1	7	
3.3	<pre>int[] weighting = {8,7,6,5,4,3,2,10,0,9,8,7,6,5,4,3,2}; PrintWriter pw = null; try { pw = new PrintWriter(new File("HotCars.txt")); } catch (FileNotFoundException ex) { JOptionPane.showMessageDialog(rootPane, "Error creating HotCars.txt file"); } for (String vin: vinArr) { int sumOfProducts = 0; char[] vinChars = vin.toCharArray(); for (int i = 0; i < vinChars.length; i++) { int numericalValue = getNumericalValue(vinChars[i]); sumOfProducts += (numericalValue * weighting[i]); } int checkDigit = sumOfProducts%11; if(checkDigit == 10){ checkDigit = 'X'; } if(vin.charAt(8) == (checkDigit+"").charAt(0)){ txaVinOK.append(vin + "\n"); }else{ pw.write(vin + "\n"); } } pw.close();</pre>	<p>Create a structure✓ to store✓ the vin character position (any other strategy to get the correct weighting) Try/catch✓ Create PrintWriter✓HotCars-file for writing✓</p> <p>Loop through vin number array✓</p> <ul style="list-style-type: none"> • Declare ✓and initialise sum variable to 0✓inside loop✓ • For each vin character✓ <ul style="list-style-type: none"> ○ Get its numerical value✓✓ ○ Get its position weighting✓ correct according to the structure used✓ ○ Multiply the two✓ ○ Add to the sum of products✓ • Mod 11✓and assign to variable✓ • Test if ✓Mod = 10 a✓and set check digit to X✓ • Obtain check digit in vin (position 9)✓ • Compare to calculated check digit✓✓ • If the same append ✓vin to text in text area✓ • If not write to HotCars.txt file✓on new line✓ <p>Close HotCars.txt file✓</p>	2 1 2 1 3 1 2 2 1 1 2 3 1 2 2 2 1	29	
Question 3 Total				43	