Western Cape Government
Education

Learner's Name: ________________________

INFORMATION TECHNOLOGY

PAPER 1 – PRACTICAL

SEPTEMBER EXAMINATION – 2014

MARKING GRID

Q1	Q2	Q3	Q4	Total
[33]	[56]	[29]	[32]	[150]

Question 1: Average Speed Prosecution – Java Programming

1.1	**private void btnChangHeadngActionPerformd()**
	lblHeading.setText("Average Speed Prosecution");
	lblHeading.setForeground(Color.white);
	pnlHeading.setBackground(Color.blue);
		Set text to label✓
		Text colour white✓
		Background colour of panel blue✓
		(Any order)
		NB: NO marks if changes in the Properties sheet, MUST be in code!
			(3)

1.2	**privte void btnCalcSpeedActionPerformed ()**
	String sTime1 = txfTime1.getText(); // in format hh:mm:ss
	String[] time1 = sTime1.split(":");
	int hr1 = Integer.parseInt(time1[0]);
	int min1 = Integer.parseInt(time1[1]);
	int sec1 = Integer.parseInt(time1[2]);
	int secT1 = (hr1 * 60 + min1) * 60 + sec1;
	String sTime2 = txfTime2.getText(); // in format hh:mm:ss
	String[] time2 = sTime2.split(":");
	int hr2 = Integer.parseInt(time2[0]);
	int min2 = Integer.parseInt(time2[1]);
	int sec2 = Integer.parseInt(time2[2]);
	int secT2 = (hr2 * 60 + min2) * 60 + sec2;
	int secDiff = secT2 - secT1;
	double hrDiff = secDiff / 3600.0;
	double distance = Double.parseDouble (txfDistance.getText());
	aveSpeed = distance / hrDiff;
	txfSpeed.setText("Average speed = " + String.format("%.2f", aveSpeed));
		reading time from correct text field✓
		extracting the hr, min and sec by ANY method(Scanner, split, substring, other)✓✓✓
		converting to seconds✓✓✓
		ONE mark for repeat of methods✓ [Mark second one if previous is wrong and this is more correct]
		subtract times (seconds)✓
		[Alternatively a more complicated method of subtracting with carry-overs (60!) can be used for the bracketed code (any correct method/result should get all 6 marks PLUS one for the repeat PLUS 1 for getting the difference in times)]
		convert seconds to a decimal fraction of hr✓
		read in km and parse✓✓
		divide distance (in km) by time (in hr)✓
		formatting 2 decimals✓
		display in correct text area✓
			(15)

1.3	**private void btnCalcFineActionPerformed()**
	String code = "";
	if (rbtSingleLane.isSelected()) { code += 'S'; }
	else { // **or** if (rbtDoubleLane.isSelected())
	code += 'D'; }
	if (rbtUrban.isSelected()) { code += 'U'; }
	else { // **or** if (rbtOutOfTown.isSelected())
	code += 'O'; }
	int speedLimit = 0;
	switch(code) {
	case "SU": speedLimit = 80; break;
	case "SO": speedLimit = 120; break;
	case "DU": speedLimit = 100; break;
	case "DO": speedLimit = 120; break; }
	int overSpeedLimit
	= (int) (aveSpeed - speedLimit);
	int factor = overSpeedLimit / 10 + 1;
	double fine = factor * 100;
	txfFine.setText(String.format("%s R%.2f", "Fine
	=", fine));
		build up code from 4 radio buttons✓✓✓✓
		set 4 speed limits according to code✓✓✓✓
		[can also use if or if..else – ANY method that works correctly]
		calculate how much over limit✓
		cast to int✓
		calculate “factor” rounded up to 10 (by adding 1)✓✓
		calculate fine by multiplication✓
		format R and 2 decimals✓
		and display✓
			(15)

	Total Question 1 =		**[33]**

Question 2: Speeding Fines Processing – Object-oriented Programming

2.1.1	**public SpeedingFine**(String registrationNumber, char laneCode, char areaCode, char roadCode, int speed)
	this.registrationNumber = registrationNumber;
	this.laneCode = laneCode;
	this.areaCode = areaCode;
	this.roadCode = roadCode;
	this.speed = speed;
		Can be done with wizard, ONE mark only✓	(1)

2.1.2	**public int getSpeedLimit() {**
	switch (areaCode) {
	case 'R':..return 60;
	case 'U': {
	if (laneCode == 'S') {
	return 60; }
	if (laneCode == 'D') {
	return 80; }
	}
	case 'P': {
	if (laneCode == 'S' && roadCode == 'N') {
	return 70; }
	if (laneCode == 'S') {
	return 80; }
	if (laneCode == 'D') {
	return 100; }
	}
	case 'O': {
	if (laneCode == 'S' && roadCode == 'N') {
	return 100; } // or one else
	if (laneCode == 'S') {
	return 120; }
	if (laneCode == 'D') {
	return 120; }
	} }
	return 1000; }
		The whole speed limit determination can of course be done with just if or if..else.
		Learners can use either an int variable for the speed limit and return that at the end, or return several times as in the example on the right.
		Either characters of the different codes can be used separately (as in the sample code on the right, or they can be joined into Strings and evaluated with the .equals method.
		E.g. if (code.equals(“DO”)
		speedLimit = 120;
		
		As there are 9 different code combinations to evaluate plus the default, the question will count 10 marks (9 plus 1 for else = 1000).
		Look out for any logic errors!
		R = always 60✓
		SU = 60✓
		SP = 80✓
		SO = 120✓
		DU = always 80✓
		DP = always 100✓
		DO = always 120✓
		SON = 100✓ Check that ‘N’ is reachable!
		SPN = 70✓ otherwise subtract 1
		Else = 1000✓
			(10)

2.1.3	**public double getFine()**
	if (!isFinable()) { return 0.0; }
	double fine = 100.0; OR int fine = 100;
	double speedDifference = speed - getSpeedLimit();
	
	int factor = 0;
	while (factor < diff / 10) { //or for-loop
	fine = fine + (factor * 200);
	factor++; }
	return fine; }
		(could be public int getFine())
		use of method isFinable()✓ (NO mark if testing again)
		return 0✓
		calculate difference in speed✓
		initialise fine and factor✓
		calculating fine with loop (while OR for) as per flowchart (4 marks)✓✓✓✓
		[Only award 2 marks if fine is determined by if..else or switch..case because it is NOT according to the given algorithm]
		returning correct value✓
			(9)

2.1.4	**public String toString()**
	if (isFinable()) {
	return String.format("%-20s%4s%12.2f%10.2f%n", registrationNumber, areaCode, getSpeed(), getFine()); }
	return String.format("%-20s%27s%n", registrationNumber,"No fine");
		test finable✓
		formatting (or tab)✓
		out: 4 elements✓
		format fine 2 decs (‘R’ optional)✓
		else or 2nd return✓
		Reg num and “No fine”✓
			(6)

2.2.1	**private void btnProcessFinesActionPerformed()**
	txaOutput.setText(String.format("%-22s%-8s%-
	8s%6s%n%n", "Registration number",
	"Type", "Speed", "Fine"));
	File f = new File("SpeedingOffences.txt");
	if(f.exists()) {
	try {
	Scanner fileReader = new Scanner(f);
	while (fileReader.hasNext()) {
	String line = fileReader.nextLine();
	Scanner lineBreaker = new
	Scanner(line).useDelimiter("#");
	String registrationNumber =
	lineBreaker.next();
	String codes = lineBreaker.next();
	char laneType = codes.charAt(0);
	char areaType = codes.charAt(1);
	char roadType = ' ';
	if (codes.length()>2) {
	roadType= codes.charAt(2); }
	int speed = lineBreaker.nextInt();
	SpeedingFine sf = new
	SpeedingFine(registrationNumber,
	laneType, areaType, roadType, speed);
	txaOutput.append(sf.toString());
	txaOutput.setCaretPosition(0); //(optional)
	} } }
	catch (Exception e) {
	JOptionPane.showMessageDialog (rootPane,
	"File reading error");
	e.printStackTrace();
	} } else {
	JOptionPane.showMessageDialog(rootPane,
	"File not found");} }
		display heading✓ getting column formatting
		(almost) right✓ [Don’t be too fussy, if tab is
		used accept a reasonable attempt]
		declaring File✓ (or file name incorporated in file
		reading object if try..catch is used)
		test if file exists (or try)✓
		declare file reading object (Scanner or
		BufferedReader)✓
		while loop to read file✓
		Inside loop:
		read one line✓
		declare line breaker object or use split(“#”)✓
		regNum: assign first string✓
		codes: assign string ✓
		and divide into chars✓
		declare a default char for road type✓ (can be any
		char NOT used)
		test whether there is a third char✓
		assign third char to roadType✓
		speed: assign int✓
		declare SpeedingFine object✓
		with correct parameters✓
		append to txa✓
		using toString✓
		error message✓
			(20)

2.2.2	***In btnProcessFinesActionPerformed() at the beginning:***
	counterFinesOtherProvinces = 0;
	totalFinesOtherProvinces = 0.0;
		In btnProcessFinesActionPerformed() at the beginning:
		initialising both attributes✓	

	In btnProcessFinesActionPerformed() in the while loop AFTER instantiating the SpeedingFine object:
	if !(sf.getRegistrationNumber().startsWith("C")
	||
	sf.getRegistrationNumber().endsWith("WP"))
		In btnProcessFinesActionPerformed() in the while loop AFTER instantiating the SpeedingFine object:
		test NOT starts with 'C'✓
		test NOT ends with 'WP'✓	

	// OR (sf.getRegistrationNumber().charAt(0) != 'C' && sf.getRegistrationNumber() .charAt(sf.getRegistrationNumber().length()-2) !='W')	correct logic✓ (AND/OR depending on NOT) [Either !(cond1 || cond2) OR !(cond1) && !(cond2)] (De Morgan's Law!)	

	{
	counterFinesOtherProvinces++;
	totalFinesOtherProvinces += sf.getFine(); }
		incrementing counter✓
		adding fine to total✓	

	In btnFinesFromOtherProvincesActionPerformed() txaOutput.setText("Fines incurred by cars from other provinces\n");	***In btnFinesFromOtherProvincesActionPerformed():*** display heading✓	
	txaOutput.append("Number = " + counterFinesOtherProvinces + "\n");	display counter✓	
	txaOutput.append ("Total fines = " + String.format("R%.2f%n", totalFinesOtherProvinces)); }	display total✓ and format fine to currency✓	(10)
		Total Question 2 =	**[56]**

Question 3: Multiple Speeding Fine Processing – Problem Solving

<table>
<tr><td>3.1</td><td><b>private void btnProcessActionPerformed()</b>
boolean[] multiple = new boolean[max];
for (int i = 0; i < multiple.length; i++) {
multiple[i] = false; }
txaOutput.setText("Offenders with multiple fines” + ”\n--------------------------\n\n");
for (int i = 0; i < max - 1; i++) {
int count = 1;

double total = arrFines[i];
for (int j = i + 1; j < max; j++) {
if (arrRegNums[i].equals(arrRegNums[j]) && multiple[j] == false) {

count++;
total += arrFines[j];
multiple[j] = true;
} }

if (count > 1) {
String out = String.format("%s%s%d%s", arrRegNums[i], " had ", count, " fines");
txaOutput.append (String.format ("%-31s%sR%.2f%n",out, "Total owing: ", total));
} }</td><td>Declaring and initialising ANY data structure, e.g. a list, an array, a text file, or an array of boolean, to assist with tracing the multiple offenders) ✓✓
Display heading✓
Loop through all entries✓
Initialise a counter✓ (to keep track of duplicates (can be replaced with any operation to keep track of multiple offenders)
Initialise total✓
Inside (nested) loop starting at i + 1✓
Testing whether regnum is equal✓
&& (AND) logic✓ ANY test whether they have been processed already✓
Increment counter✓
Add fine to total✓
Set Boolean to true to indicate that this record has been processed or add to the data structure declared above✓
Test count > 1 indicating multiple fines✓ (or check whether in data structure of duplicates)
Display information reasonably formatted (columns NOT required)✓✓</td><td>(16)</td></tr>
<tr><td>3.2</td><td><b>private void btnBirthdayActionPerformed()</b>
Random random = new Random();
int ranNum = 0;
do {
ranNum = random.nextInt(76); //or use max
} while (arrFines[ranNum] != 100.0
|| multiple[ranNum] == true);
JLabel message = new JLabel();
pnlBirthdayMessage.add(message);
message.setBounds(30, 10, 500, 30);
message.setText(" Happy Birthday Chief! "
+ arrRegNums[ranNum]
+" will not have to pay his fine!");
message.setFont(new Font("Papyrus", 3, 12));
message.setBackground(Color.red);
message.setForeground(Color.white);
message.setOpaque(true);
btnBirthday.setEnabled(false);</td><td>ANY random method used correctly✓
initialise number outside loop✓
do..while loop to repeat✓
randomise between 0 and 76✓
fine must be R100.00 (not more)✓
must not be one of the multiple offenders✓
dynamic instantiation of label✓
add label to panel✓
setBounds as instructed✓
message text as instructed✓
add registration number as randomised into message✓

any ONE formatting feature✓ (ONE mark only!)

disabling Birthday button✓</td><td>(13)</td></tr>
<tr><td></td><td colspan="2"><b>Total Question 3 =</b></td><td><b>[29]</b></td></tr>
</table>

Question 4: Traffic Fine Statistics – Problem-Solving

4.1	**private void btnDisplayActionPerformed()**
	txaOutput.setText("");
	for (int i = 0; i < headings.length; i++) {
	txaOutput.append(headings[i] + "\t");
	if (headings[i].length() < 8) {
	txaOutput.append("\t");
	} }
	txaOutput.append("\n------------------------------------
	----------------------------------\n");
	for (int i = 0; i < stats.length; i++) {
	txaOutput.append(towns[i] + "\t");
	if (towns[i].length() < 8) {
	txaOutput.append("\t"); }
	for (int j = 0; j < stats[i].length; j++) {
	txaOutput.append(stats[i][j] + "\t\t"); }
	txaOutput.append(totals[i] + "\n"); }
		
		clear txa✓
		loop to display headings✓ (or concatenation)
		headings appended with \t (or other)✓
		extra spaces for small headings✓
		
		long line✓
		
		outer loop to 6✓
		append towns' names✓
		extra spaces for small towns✓
		
		inner loop to 3✓ (or concatenation)
		append numbers✓ (or concatenation)
		append totals✓
		plus new line✓
			(12)

4.2	**private void btnTotalsActionPerformed()**
	for (int i = 0; i < totals.length; i++) {
	totals[i] = 0; }
	for (int i = 0; i < stats.length; i++) {
	for (int j = 0; j < stats[i].length; j++) {
	totals[i] += stats[i][j]; } }
		
		initialise totals[] to 0✓
		
		outer loop to 6✓
		inner loop to 3✓ (or concatenation)
		add correct number✓ (or concatenation)
		to correct total✓
			(5)

4.3	**private void btnHighestActionPerformed()**
	int xIndexHighest = 0;
	int yIndexHighest = 0;
	for (int i = 0; i < stats.length; i++) {
	for (int j = 0; j < stats[i].length; j++) {
	if (stats[i][j] >
	stats[xIndexHighest][yIndexHighest]) {
	xIndexHighest = i;
	yIndexHighest = j;
	} } }
	txaOutput.append("\n\nThe highest monthly is
	R" + stats[xIndexHighest][yIndexHighest]
	+ " recorded in " + towns[xIndexHighest]
	+ " for "+ headings[yIndexHighest+1]);
		
		Initialise variables to store highest, town, month
		or use indexes✓✓
		outer loop to 6✓
		inner loop to 3✓
		test highest against current value✓
		if higher, store details✓
		
		build up string with value, town and month✓
		append to txa✓
			(8)

4.4	**privat void btnAddAprilFiguresActionPerformd()**
	int[] april = {562873, 342126, 23419,
	156321, 243111, 101345};
	headings[1] = headings[2];
	headings[2] = headings[3];
	headings[3] = "April";
	for (int i = 0; i < stats.length; i++) {
	stats[i][0] = stats[i][1];
	stats[i][1] = stats[i][2];
	stats[i][2] = april[i]; }
		
		new array for April's figures✓✓
		
		move every month up one position (can also be
		done in loop)✓
		add "April" at end✓
		loop to 6✓
		
		move every figure up one position✓
		add April's figure✓
			(7)

		Total Question 4 =	**[32]**