MACARTHUR ANGLICAN SCHOOL

Established traditions, proven success!

Technology & Creative Design Faculty
11 Software Design & Development

Task Name: Social & Ethical Issues Case study

Weighting: 35%

Due Date: 17 June 2016

Outcomes:

- P1.1 Describes the functions of hardware and software
- P1.2 Describes and uses appropriate data types
- P1.3 Describes the interactions between the elements of a computer system
- P3.1 Identifies the issues relating to the use of software solutions
- P4.1 Analyse a given problem in order to generate a computer-based solution

Description of Task:

Read the attached case study on how speed cameras work in NSW.

Create a data flow diagram to represent the system based on the information from the article. You may wish to break down more complicated parts of the system into smaller data flow diagrams.

You are also given a Safe-T-Cam problem. You are to create an IPO chart in your quest to define and understand the problem. You will also need to apply the top-down design process to solve the problem. That is, you need to show a diagram that has the problem broken down into smaller, more manageable problems (i.e. a structure chart). You will also need to write the pseudocode for the problem and finally submit the JavaScript code for the program.

For Submission:

- • Data flow diagram(s)
- • IPO chart for the Safe-T-Cam problem
- • Structure chart for Safe-T-Cam problem
- • Pseudocode for the Safe-T-Cam problem
- • JavaScript code

Assessing Teacher: B Hall

Date Handed Out: 29 April 2016

Penalties for Late Submission:

Penalties for late submissions are as follows:

- 1 day late - 30% subtracted from the marks awarded
- 2 days late - 60% subtracted from the marks awarded
- 3 or more days late - 100% subtracted from the marks awarded (zero marks awarded)

How speed cameras work in NSW

Image courtesy of :http://www.thecurrentmagazine.com/2011/04/speedcameras-ripping-off-ripped-out/

Speed cameras detect the speed of vehicles by using detectors embedded into the road surface or radar technology. Red-light speed cameras also detect vehicles travelling over the stop line or entering the intersection after the lights have turned red. If the speed of the vehicle exceeds the legal limit, or a vehicle runs a red light, a digital picture is taken of the offending vehicle.

The image clearly shows the colour, type, make and number plate of the vehicle. Digital images also include:

- Date of the offence
- Time of the offence
- Location details of the camera that took the picture
- Direction of travel of the offending vehicle
- Speed of the offending vehicle
- Speed limit on the road where the camera is positioned
- The lane that the vehicle was travelling within
- Other security and integrity parameters

The original recorded images are stored electronically. A security indicator is also produced when the file is written to prevent tampering with the image. When the image is electronically transferred, all images and relevant information (such as time, date and location) are encrypted.

All measurements taken by the cameras in NSW are traceable to national standards. The Roads and Maritime Services (RMS) time server is synchronised every 30 minutes using the Network Time Protocol to the National Measurement Institute's time service. The National Measurement Institute's time service provides the time standard for all of Australia. RMS has procedures in place where a central computer confirms the synchronisation of the enforcement camera time against Telstra time every 30 minutes and alerts if this synchronisation does not exist.

Once the images have been uploaded to a central server, the State Debt Recovery Office (SDRO) system issues a penalty notice which gets mailed to the offender. If the offender appeals the penalty and wishes to go to court, this will be recorded in the database. A court date will be set and details of the appeal will be mailed to the offender.

When an offender pays the fine, the offender's record is updated on the server and an invoice is generated and emailed to their email address.

Disclaimer: Most of the above article was taken from http://roadsafety.transport.nsw.gov.au/speeding/speedcameras/howdo_theywork.html
The way that fines are processed have been described based on anecdotal experience and may not accurately represent the actual NSW system.

MACARTHUR ANGLICAN SCHOOL
Established traditions, proven success!

Safe-T-Cam Problem

Drivers slow down briefly for marked speed cameras, but calculating average speeds between checkpoints can be a more effective way of detecting consistently speeding vehicles.

Assume there are three checkpoints numbered 1, 2 and 3, spaced along a highway with a 110km/hr speed limit. Checkpoints 1 and 2 are 133.0 km apart and checkpoints 2 and 3 are 57.5km apart. Each checkpoint has an accurate clock and reliable numberplate recognition equipment. After each vehicle passes, the time, checkpoint number and numberplate are sent to a central log.

Your task is to read the log and identify any speeding vehicles.

The data is presented as a single file with the rows in strict time order, preceded by the number of rows on the first line (up to 300). Each data item consists of three fields, separated by a space:

- A time stamp in 24-hour format hh:mm:ss. All three components have 2 digits. Times are all within the same day.
- The checkpoint number (1, 2 or 3)
- The vehicle's number plate, up to 6 alphanumeric characters.

You can assume the supplied data uses the correct format, all times are legitimate and so on.
The program's output consists of only the rows that indicate that a vehicle's average speed has exceeded the speed limit for the section that ends at that checkpoint, along with the calculated average speed in km/hr to 1 decimal place.
The output must be in chronological order, same as the input.

Example

Input:

```
7
02:12:49 1 MYTRUK
02:21:52 1 BI66BT
03:17:21 2 MYTRUK
03:22:55 2 2SLOW
03:42:18 2 BI66BT
03:55:37 3 2SLOW
04:12:58 3 BI66BT
```

Output:

```
03:17:21 2 MYTRUK 123.7
04:12:58 3 BI66BT 112.5
```

Test Data

You should test your program on the following long data set:
Refer to the Safe-T-Cam test data file on Moodle

Marking Rubric

Components	Developing (0-1)	Satisfactory (2-3)	Substantial (4-5)
Data flow diagram			
Accuracy	Limited data flow diagram submitted.	Data flow diagram mostly shows the correct use of symbols and semantics.	Data flow diagram shows correct use of symbols and semantics, with little or no mistakes.
Completeness	Data flow diagram only depicts part of the system described.	Data flow diagram clearly depicts the system described in the case study. There is at least one instance of a Level 2 DFD.	Data flow diagram thoroughly depicts the system described in the case study. There are at least two instances of a Level 2 DFD and possibly a Level 3 DFD.
Structure chart			
Top-down design	Structure chart shows limited breakdown of the problem into smaller problems.	Structure chart clearly shows the break down of the Safe-T-Cam problem into smaller, more manageable modules.	Structure chart thoroughly shows the break down of the Safe-T-Cam problem into smaller, more manageable modules.
Accuracy	Limited structure chart submitted.	Structure chart mostly shows the correct use of symbols and semantics.	Structure chart shows correct use of symbols and semantics, with little or no mistakes.
IPO Chart			
Accuracy	Limited IPO Chart submitted	IPO chart clearly shows the inputs, processes and outputs to solve the Safe-T-Cam problem.	IPO chart thoroughly shows the inputs, processes and outputs to solve the Safe-T-Cam problem.
Pseudocode			
Syntax	Pseudocode syntax have not been followed.	For the most part, pseudocode syntax has been followed. That is: • keywords are written in capitals • structural elements come in pairs, eg for every BEGIN there is an END, for every IF there is an ENDIF • indenting is used to identify control structures in the algorithm • the names of subprograms are underlined.	Pseudocode syntax has been followed for all pseudocode handed in. That is: • keywords are written in capitals • structural elements come in pairs, eg for every BEGIN there is an END, for every IF there is an ENDIF • indenting is used to identify control structures in the algorithm • the names of subprograms are underlined.

Components	Developing (0-1)	Satisfactory (2-3)	Substantial (4-5)
Python code			
Intrinsic documentation	Limited use of comments, meaningful variable names and white space	At least two of the following: comments, meaningful variable names and white space, have been used in the code.	Comments, meaningful variable names and white space have been used in the code.
Robustness	The code does not compile	The code compiles but does not completely solve the Safe-T-Cam problem.	The code compiles and solves the Safe-T-Cam problem. The test data provided is successfully processed and produces the correct output.

Feedback:

____

____

____

Total Marks: /40