

Coursework: Data Viewer

Component	Times and Dates	Weighting (as a proportion of coursework)
1. Full Program (cd or floppy disk) + Documentation (paper)	Office at Sopwith Building reception 1pm, Friday March 27th	50%
2. Test/Demo	If required to give a demo, you will be contacted by your university e-mail.	This effects the interpretation of the above grades

Please Note:

You must meet all deadlines set. Failure to do so will result in a penalty. The usual deadline time is 1pm on the stated day – ALL work received after this time will be stamped LATE by Student Office staff.

Work submitted late but within a week of the deadline will be capped at 50% and receive a maximum grade of Pass (-) . Work submitted beyond a week of the deadline without approval will get 0% with a grade of F0.

*If, however, you have a serious problem which prevents you from meeting the deadline you may be able to negotiate an extension **in advance**. In the first instance you should contact the Student Liaison Officer, Holly Rook in the Student Office for advice. However any extension will need to be formally agreed by your Module Leader who will liaise with Holly to confirm and agree a new hand in date. Your work will then be marked without penalty.'*

You may be required to give a demonstration of your program or complete a coursework test. If this is the case then failure to do so will result in a zero mark. The demonstration/test is designed to test your understanding of your work and will affect the marks and grades for all of your components.

The work you submit must be your own, independent work. Submitting other people's work as if it were your own is plagiarism. Code written by anyone else other than yourself must be separated from your code and highlighted IN THE CODE LISTING AND properly acknowledged IN YOUR DOCUMENTATION.

All code will be analysed by plagiarism detection software, any matches with work from this year or previous years will then be inspected and penalised depending upon the severity of the plagiarism. At worst you will receive no marks and be subject to the University plagiarism procedure.

As you work, keep the last working version intact and continue to improve another version, and keep them both backed-up by having at least one copy on your network H:drive!

1. OBJECTIVES, the DATA VIEWER and CRITERIA

The objective of the coursework is for you to develop a Data Viewer. The Data Viewer should be an application that lets a user preview some raw data, perform some calculations on that data, and display the results of the calculations as text and as an appropriate graph or chart. You will be allocated some suitable data for the purpose (this is described in an accompanying document). The data that you are allocated will be a subset of the dataset that is unique to you. You will also be allocated some user interface requirements that are unique to you. All other design decisions are up to you.

As you will see, there are many ways of interpreting the specification. There is no single 'correct' program. 'Easy' interpretations (involving less processing of information and/or coding) earn fewer marks than more 'difficult' interpretations. Your aim is to explore and understand important Java concepts and classes. Some of these are listed below. How have/could you use these concepts and classes in your Data Viewer?

Data Types

Logical Operators

Control Structures

Arrays

Swing Components (ListBoxes, LayoutManagers)

Drawing

Various Media (sound, images etc.)

Inheritance

Encapsulation

Polymorphism

Input/Output Streams

Multi-Threading

Event Handling

Exception Handling

Submission Checklist

1. An *electronic* copy (on CD or floppy disk) of working program with a graphical user interface that allows a user to:
 - Load data from a file
 - View the raw data
 - Calculate the required summary information from the raw data
 - View the summary information as text
 - Plot summary information as a chart
 - any additional, special functions of your choice

The program must be submitted as .java and .txt files, and optionally, as also a .jar file and **NetBeans v6** project. The program should run on a machine using a standard Java **JDK version 1.6** install. If your program has any other requirements (e.g. data files or other .jar files) then this must be included and instructions of how to set them up included in your user guide (see 3.)

2. An *electronic* copy of Class specifications (specifications of your classes, their methods and attributes in a standard format generated by the JavaDoc tool)
3. A *paper* print out of a complete ‘walk-through’ of your program from the user’s point of view i.e. step-by-step screen dumps, annotated with user actions (“Click here to ...” etc). (hard copy). The walkthrough should begin with instructions for how to run your program.
4. A *paper* print out of a UML diagram of your program, indicating classes, methods, variables, and their relationships
5. Zipped folder with all files (program, documentation, docs) submitted via StudySpace. A link will be provided.

3. Your Personal Data Viewer

Each student implements a data viewer that is unique to that student. The unique features of ‘Your Data Viewer’ are defined in terms of the data viewed, and the user interface components used. During the week of the module, your personal specification will be e-mailed to you. Till then, here are the range of options from which we will select your particular specification.

Data

The data we are using for this coursework is an extract of an actual survey of the way students use mobile telephones during a rendezvous (meeting for lunch, going to the cinema etc..). The actual data values are of no interest to this module, so do not worry about what the survey results do or do not mean – it is just more realistic to work with actual data than ‘contrived’, or fictitious data.

The data comprises 3 types of variable:

- **selection variables:** for filtering out unwanted data i.e. selecting a subset of the data to perform subsequent calculations on. Example selection variables are: year in which data collected, the month in which data was collected, the course the student was enrolled on (under-graduate or post-grad), or marital status. Let’s say, for this example calculation, we are only interested in rendezvous reported in February.
- **X-axis, or condition variables:** under what conditions were the data gathered? when the results of a calculation is displayed, this variable is plotted horizontally. Example X-axis variables are: sex, age (as a category e.g. 18-25 or 26-45), rendezvous context (before or en route to the meeting). Let’s say we want to compare different age groups.
- **Y-axis, or user performance variables:** what aspect of user behaviour might vary depending on the conditions? When the results of a calculation is displayed, this variable is plotted vertically. Example Y-axis variables are: success (arrived on time or late), usage of communication services, rated ease of use. Let’s say we want to look at the relative use of communication services – perhaps 18-25s use text messaging more than 26-45s.

Your program selects a subset of data to view according to values of the selection variable, calculates the frequency of each category of user performance for each

condition, and then draws a graph of the results (with condition variable on the X-axis and user performance variable on the Y-axis).

Selection Variable

- A: Year in which data reported (2001 or 2002)
- B: Month in which data reported (February or March)
- C: Course the student is enrolled on (undergraduate or post graduate)
- D: Marital status (married, same partner for >1 yr, single)

X/Condition Variable

- A: Sex (male or female)
- B: Age (18-25s and 26-45s)
- C: Met at Rendezvous Point Before (yes or no)

Y/User Performance Variable

- A: Success (yes or no)
- B: Stress (yes or no)
- C: Lost an Opportunity (yes or no)

Special Function

Enhance the basic program with additional functionality of your choice, but of the following kinds: Editing data; Searching for data; Summarising data.

User Interface Options

Design Pattern

- A. Tabbed Dialog e.g. property inspectors
- B. Multiple Windows e.g. MS Office

See examples in many Microsoft applications.

Control for setting value of selection variable

- A. Radio buttons
- B. Drop-down list box
- C. Check boxes

3 settings are implied – value 1 (e.g. year = 2001, gender = male etc.), value 2 (year = 2002, gender = female etc.) and both

Type of Chart

- A. Column
- B. Pie

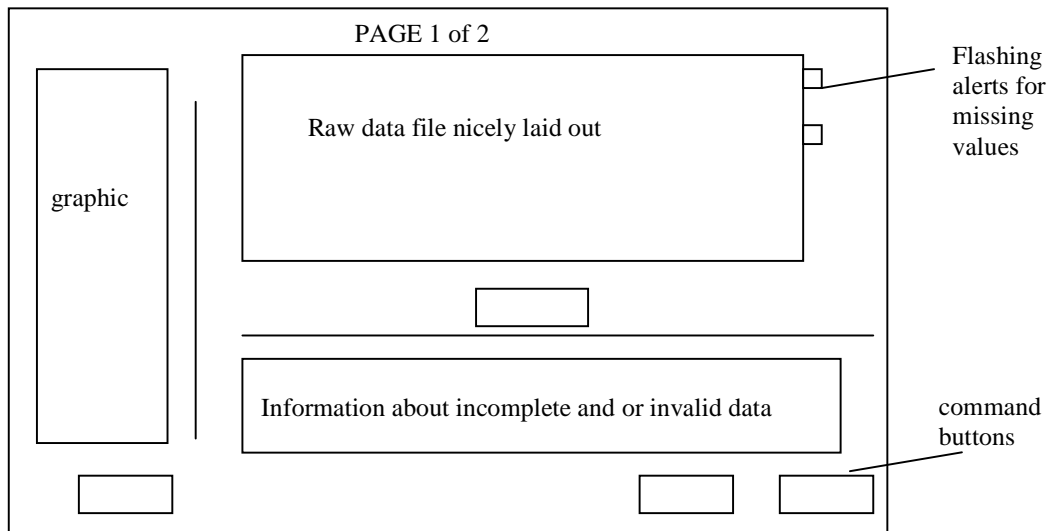
For column or bar charts, the calculation is a frequency e.g. how many were successful, or unsuccessful. For pie charts, the calculation is a proportion e.g. what proportion of rendezvous were successful?

Example Personal Specification

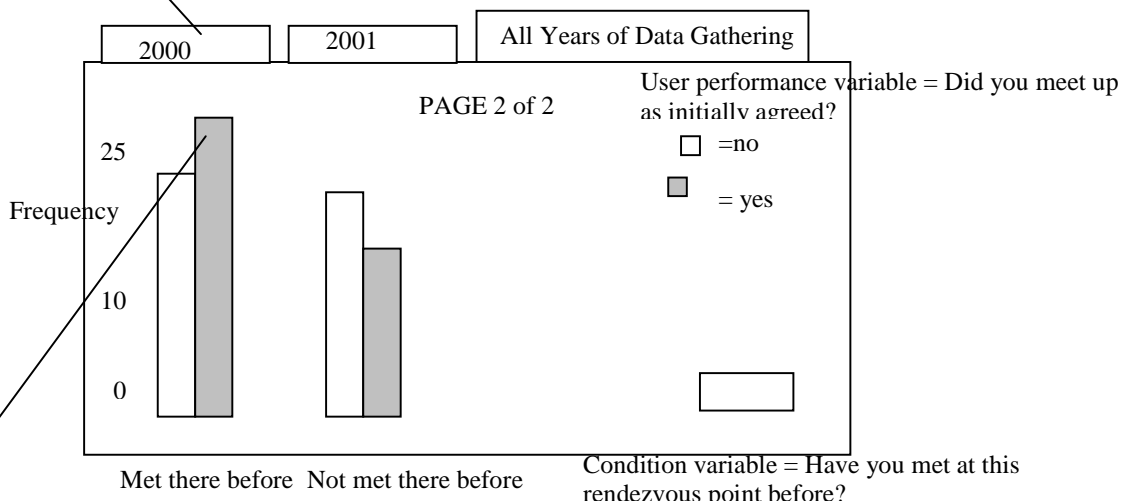
- Selection Variable: year of reporting (2001 or 2002)
- X/Condition variable: met at that rendezvous point before or not
- Y/User Performance variable: met up as agreed, or not
- UI Pattern: Multiple Windows
- Selection Variable Controls: tabs
- Chart Type: bar

The data viewer for the above option might resemble the figure below:

Fig. 1: Example Data Viewer Rough Sketch (for illustration, only)



Selection variable = year of reporting



So, the example chart shows that every year data was collected, we found that people were more likely to meet up as agreed, if they had previously met at that rendezvous point.

Marking Scheme

Name:

ID: K.

	Value	Comments	Mark
Compilation (when the Project is opened in NetBeans5, does it compile and run without further intervention?)	10		
Data Model (e.g. are objects and their relations correctly represented using e.g. Classes and ArrayLists)	10		
Graphical User Interface (does it comply with look and feel guidelines? Is it intuitive to use? Responds to user actions?)	10		
Performs Calculations (correct? Output to console/displayed?)	10		
Displays Chart (laid out clearly?)	10		
Special Functionality (appropriate? Imaginative?)	10		
Reads Data From a File (changes to values in text file e.g. from 'y' to 'n', or from 'y' to ' ' are handled and displayed)	10		
JavaDoc (are classes, methods and fields usefully described?)	10		
UML diagram (complete? clearly laid out? Modular design for re-use? Hard copy?)	10		
Walkthrough (complete? Accurate? Bugs/undesirable behaviour identified?)	10		
Total	100		