

DLTV Resource Kit

For use with the VCE Applied Computing 2020–2024 Study Design

|  |
| --- |
| Applied Computing: Software Development Unit 3 Area of Study 1 Software development: Programming School-assessed Coursework: Alternate task |

**Copyright:** All Victorian Curriculum and Assessment Authority (VCAA) material is copyright. Area of study outcomes, key knowledge, key skills and assessment task descriptions reproduced by permission from the *VCE Applied Computing* (2020–2024) © Victorian Curriculum and Assessment Authority (VCAA). VCE is a registered trademark of the VCAA. The VCAA makes no warranties regarding the correctness or accuracy of this DLTV resource. Teachers are advised to check the [VCAA Bulletin](http://www.vcaa.vic.edu.au/Pages/correspondence/index.aspx) for updates. The current VCE study design and related content can be accessed directly at [www.vcaa.vic.edu.au](http://www.vcaa.vic.edu.au/) and has been reproduced with permission.

**Disclaimer:** This resource has been written by the author for Digital Learning and Teaching Victoria (DLTV), for inclusion in their DLTV Resource Kit 2020–2024 to be utilised by teachers and students of VCE Applied Computing. This does not imply that it has been endorsed by the Victorian Curriculum and Assessment Authority (VCAA).

While every care has been taken by the author, they accept no responsibility for the accuracy or advice contained in this task. Teachers are asked to preview and evaluate all DLTV Resource Kit resources before using them or distributing them to students.

This assessment task is intended as a sample only. Please note the following from the VCE and VCAL Administrative Handbook 2019:

'Where commercially produced tasks are being used for School-based Assessment, the school should ensure the tasks meet the requirements of the study design and that they have been sufficiently modified to enable student work to be authenticated.’

*—*[*VCE and VCAL Administrative Handbook 2019*](https://www.vcaa.vic.edu.au/Documents/handbook/2019/adhb19.pdf), VCAA, 2019, page 74

# VCE Applied Computing: Software Development

## Unit 3 Area of Study 1: Programming practice

On completion of this unit the student should be able to interpret teacher-provided solution requirements and designs, and apply a range of functions and techniques using a programming language to develop and test working software modules.

Students apply computational thinking skills when interpreting given solution requirements and designs, and when developing them into working modules.

**Task conditions**

**Allowed resources:** *[to be determined by the teacher]*

**Time allocated to this task:** *[to be determined by the teacher]*

**Marks allocated:** *100 marks*

## Outcome brief

A group of students at Galactic Senior Secondary School are setting up a second-hand shop to buy and sell used textbooks. At the end of the year, older students can sell their used textbooks to the shop, and the shop will make a small profit when selling them to younger students for the following year.

They have approached you to develop simple software to assist with their shop.

## General information

You must complete four tasks of increasing complexity by writing programs to implement the design provided. The designs in all cases are provided for you, however, keep in mind that you must:

* demonstrate the use of functions, with parameters and return values, where appropriate.
* include rich internal documentation that fully describes your logic and reasoning and implement code layout conventions (e.g. white space, indenting) for readability.
* use meaningful names for your project, data structures, functions and variables, adhering, where reasonable, to code conventions for the programming language.
* validate input data and user input.
* fully and formally test your programs, providing clear evidence of operation.

**Testing**

You should perform testing for each task. For Tasks 2, 3 and 4 this includes constructing appropriate test data based on the example.

A testing table, like the template provided in Appendix 1, should be provided as evidence.

**Task 1**

**What are my old textbooks worth?** (sequence, branching, iteration, variables, functions)

The first program, to be used at the end of a year, allows a student to determine how much they will get for their collection of used textbooks if they sell it to the shop. It should work as follows:

STEP 1: Prompt the user to input the following for a textbook:

* how old is the textbook (for how many years has it been used)?
* how much did you pay the textbook (at the start of the year)?

STEP 2: Calculate and display the current value of the textbook. This is what the shop will pay for it. See the next page for details on how to calculate this value.

STEP 3: Display the **cumulative** value of the collection of textbooks based on what has already been entered.

STEP 4: Ask the user if they wish to enter another textbook. If yes, go back to STEP 1. If no, end the program.

|  |
| --- |
| How old is the textbook (in years)? **3**  How much did you pay for the textbook? **110**  This textbook is worth $44.00  The collection so far is worth $44.00  Do you wish to enter another textbook? (y or n) **y**  How old is the textbook (in years)? **1**  How much did you pay for the textbook? **95**  This textbook is worth $76.00  The collection so far is worth $120.00  Do you wish to enter another textbook? (y or n) **n** |

*Mock-up of program in use. User inputs in* ***bold****.*

**The current value of a textbook is calculated as follows:**

* the textbook loses 20% of its original value when purchased after one year.
* across the second year, it is devalued more. It loses 40% of its value when purchased at the start of that year.
* across the third year, it will lose 60% of its value from the beginning of the year, and so on.
* all textbooks become worthless after five years.

##### **Pseudocode**

This pseudocode describes a function to perform the calculation:

**Function** calculateCurrentValue(purchasedValue, age)

depreciation ← purchasedValue × 0.2 × age

**If** depreciation > purchasedValue

**Return** 0

**End If**

**Return** purchasedValue - depreciation

**End Function**

## Task 2

## Calculate profit (reading in CSV file)

It's the end of the year and the shop has bought and sold over 100 books. Now the students running the shop want to know how much profit they made.

The table printed in Appendix 2 gives an example of the data file recording each sale.

The fields (columns) are described as follows:

* **textbook** – the name of the textbook that was bought and sold
* **subject** – the school subject that requires that textbook
* **seller** – the name of the student who sold the book to the shop. *(Note: a seller may appear more than once)*
* **purchase price** – the value paid by the shop to the original owner
* **purchaser** – the name of the student who bought the book from the shop, or 'NA' if the book never got sold *(Note: a purchaser may appear more than once)*
* **sale price** – the value paid to the shop by the new owner, or 'NA' if the book never got sold
* **rating** – currently unused, but will be used in the next task.

This program should work as follows:

STEP 1: Read in the CSV file row by row, keeping a cumulative total profit:

* if the book was sold, calculate the profit for that book by subtracting the **purchase price** from the **sale price**. This value is added to the cumulative total profit.
* if the book never got sold, it is a loss and its **purchase price** must be subtracted from the cumulative total profit.
* display the book name and profit or loss for that book.

STEP 2: When the end of the file is reached, display the total profit.

##### **Pseudocode**

This pseudocode describes the operation of the program (without validation added).

totalProfit ← 0

**For** each line in the csv file

textbook ← first field from line

purchasePrice ← fourth field from line

salePrice ← fifth field from line

**If** salePrice = 'NA'

profit← purchasePrice× -1

**Else**

profit← salePrice - purchasePrice

**End If**

totalProfit ← totalProfit + profit

**End For**

**Display** 'Total profit is ', totalProfit

## Task 3

**Rate a book** (writing a CSV file, data structures, searching algorithms)

The students want to set up a kiosk in the shop so that buyers can leave a star rating for the book they bought. (Note, they will not rate the content, but the state of the book purchased, e.g. wear and tear). The program will allow a user to search for their purchase, then add or replace a star rating.

You'll start with the same CSV file used in Task 2 (also printed as a table in Appendix 2), which is an example of the data file recording each sale.

The fields (columns) are described as follows:

* **textbook** – the name of the textbook that was purchased and sold
* **subject** – the school subject which requires that textbook
* **seller** – the name of the student who sold the book to the shop. *(Note: a seller may appear more than once)*
* **purchase price** – the value paid by the shop to the original owner
* **purchaser** – the name of the student who bought the book from the shop, or 'NA' if the book never got sold. *(Note: a purchaser may appear more than once)*
* **sale price** – the value paid to the shop by the new owner, or 'NA' if the book never got sold
* **rating** – a whole number between 1 and 5 representing a star rating, or 'none' if the book hasn't been rated yet.

**This program should work as follows:**

STEP 1: Read in the CSV file *once only*. For each row in the file, store the values in a record with appropriate fields to match the columns. All the records should be kept in a list.

STEP 2: Prompt the user to enter:

* the name of the book.
* the first name of the purchaser.
* the surname of the purchaser.

STEP 3: Find the matching record in the list by searching for the name of the book in the textbook field *AND* the complete name of the purchaser in the Purchaser field. The search should be case insensitive, but otherwise the text for both fields must be an exact match.

* if a matching record is found, display it and go to STEP 4

If a matching record is not found, display an appropriate response and return to STEP 2.

STEP 4: Prompt the user to enter a new star rating from 1 to 5.

STEP 5: Update the record with the new star rating, replacing any existing value for the rating field.

STEP 6: Write all the records into a new output CSV file. The output CSV should be valid CSV, able to be opened in spreadsheet software and using the same columns as the input CSV file.

STEP 7: End the program.

|  |
| --- |
| Loading file...  Done.  Let's find your entry.  Enter the textbook name: **mapping our quadrant**  Enter your full name: **donald trump**  Sorry, there was no match. Let's try again.  Enter the textbook name: **mapping our quadrant**  Enter your full name: **camie giefer**  Found it!  What rating will you give (1 to 5)? **3**  New rating has been applied. Saving output file. |

*Mock-up of program in use. User inputs in* ***bold****.*

##### **Pseudocode**

This pseudocode describes the operation of the program (without validation added).

**Display** 'Loading file…'

**For each line in the csv file**

entry.textbook ← first field from line

entry.subject ← second field from line

entry.seller ← third field from line

entry.purchasePrice ← fourth field from line

entry.salePrice ← fifth field from line

entry.purchaser ← sixth field from line

entry.rating ← seventh field from line

**Add** entry **to list** entries

**End For**

**Display** 'Done.'

**Display** "Let's find your entry."

**While we haven't found the matching entry in list entries**

textbookToFind ← **Input** 'Enter the textbook name: '

purchaserToFind ← **Input** 'Enter your full name: '

**Search** entries **for a matching entry with** textbookToFind **and** purchaserToFind

**If we haven't found the matching entry in list** entries

**Display** "Sorry, there was no match. Let's try again."

**End if**

**End while**

newRating← **Input** 'What rating will you give (1 to 5)? '

matchingEntry.rating ← newRating

**Display** 'New rating has been applied. Saving output file.'

**Write** CSV header into output file

**For each** entry **in** entries

**Write** entry fields separated by commas into output file

**End for**

## Task 4

## Filtering and sorting the sale data (sorting algorithms)

Finally, the students would like a simple software solution to help examine their sale data. They'd like a menu of options to:

* display the data only for a specific subject.
* display the data only for a specific textbook.
* sort the data by rating.

You'll start with the same CSV file used in TASKS 2 and 3 (also printed as a table in Appendix 2), which is an example of the data file recording each sale.

The fields (columns) are described as follows:

* **textbook** – the name of the textbook that was bought and sold
* **subject** – the school subject which requires that textbook
* **seller** – the name of the student who sold the book to the shop *(Note: a seller may appear more than once)*
* **purchase price** – the value paid by the shop to the original owner
* **purchaser** – the name of the student who bought the book from the shop, or 'NA' if the book never got sold *(Note: a purchaser may appear more than once)*
* **sale price** – the value paid to the shop by the new owner, or 'NA' if the book never got sold
* **rating** – a whole number between 1 and 5 representing a star rating, or 'none' if the book hasn't been rated yet.

**This program should work as follows:**

STEP 1: Read in the CSV file *once only*. For each row in the file, store the values in a record with appropriate fields to match the columns. All the records should be kept in a list.

STEP 2: Prompt the user to enter one of the following commands:

* 'subject',
* 'textbook',
* 'rating',
* 'quit'.

STEP 3: Based on the user's choice:

* if the user entered 'subject', prompt the user for a subject name to use as a filter. Then, display only entries whose **subject** contains the response. *Note, an exact match is not required. Part of the text is allowed. eg. ‘year 10’ should bring include all subjects containing ‘year 10’, case insensitive.*
* if the user entered 'textbook', prompt the user for a textbook name to use as a filter. Then, display only entries whose **textbook** contains the response. *Note, an exact match is not required. Part of the text is allowed. eg. ‘ninja’ should bring include all textbooks containing ‘ninja’, case insensitive.*
* if the user entered 'rating', display all entries sorted by **Rating** in ascending order, with 'none' entries last.
* if the user entered 'quit', end the program now.

STEP 4: Return to the menu at STEP 2.

|  |
| --- |
| MAIN MENU  Enter 'subject' to filter list by subject.  Enter 'textbook' to filter list by textbook name.  Enter 'rating' show complete list sorted by rating.  Enter 'quit' to end program.  > **subject**  FILTER BY SUBJECT  Enter the subject to use as filter: **year 10**  Understanding silicon life (Year 10 Aliens) $63.20 -> $75.00 RATING: none  First contact: a guide (Year 10 Aliens) $107.40 -> $130.00 RATING: none  First contact: a guide (Year 10 Aliens) $107.40 -> $120.00 RATING: 5  Understanding silicon life (Year 10 Aliens) $63.20 -> $75.00 RATING: 5  Lifeforms and societies(Year 10 Planets) $39.60 -> $50.00 RATING: 3  MAIN MENU  Enter 'subject' to filter list by subject.  Enter 'textbook' to filter list by textbook name.  Enter 'rating' show complete list sorted by rating.  Enter 'quit' to end program.  > **textbook**  FILTER BY TEXTBOOK NAME  Enter the textbook name to use as filter: **ninja**  Understanding silicon life (Year 10 Aliens) $63.20 -> $75.00 RATING: none  Mapping our quadrant (Year 9 Navigation) $21.80 -> $45.00 RATING: none  Understanding silicon life (Year 10 Aliens) $63.20 -> $75.00 RATING: 5  Mapping our quadrant (Year 9 Navigation) $65.40 -> $90.00 RATING: none  Understanding silicon life (Year 10 Aliens) $47.40 -> $70.00 RATING: none  MAIN MENU  Enter 'subject' to filter list by subject.  Enter 'textbook' to filter list by textbook name.  Enter 'rating' show complete list sorted by rating.  Enter 'quit' to end program.  > **rating**  SORT BY RATING  Understanding silicon life (Year 10 Aliens) $63.20 -> $75.00 RATING: 0  First contact: a guide (Year 10 Aliens) $107.40 -> $130.00 RATING: 0  Mapping our quadrant (Year 9 Navigation) $43.60 -> $55.00 RATING: 0  Faster than light propulsion(Year 11 Engines) $51.60 -> $75.00 RATING: 0  Understanding silicon life (Year 10 Aliens) $31.60 -> $45.00 RATING: 1  Planets and satellites (Year 9 Planets) $23.80 -> $40.00 RATING: 2  Planets and satellites (Year 9 Planets) $95.20 -> $105.00 RATING: 2  Lifeforms and societies(Year 10 Planets) $39.60 -> $50.00 RATING: 3  First contact: a guide (Year 10 Aliens) $35.80 -> $45.00 RATING: 3  Mapping our quadrant (Year 9 Navigation) $21.80 -> $45.00 RATING: 3  First contact: a guide (Year 10 Aliens) $107.40 -> $125.00 RATING: 4  Faster than light propulsion(Year 11 Engines) $25.80 -> $35.00 RATING: 4  Safety on molten worlds (Year 9 Planets) $89.40 -> $115.00 RATING: 4  First contact: a guide (Year 10 Aliens) $107.40 -> $120.00 RATING: 5  Lifeforms and societies(Year 10 Planets) $39.60 -> $NA RATING: 5  First contact: a guide (Year 10 Aliens) $71.60 -> $95.00 RATING: 5  Faster than light propulsion(Year 11 Engines) $51.60 -> $75.00 RATING: 5  MAIN MENU  Enter 'subject' to filter list by subject.  Enter 'textbook' to filter list by textbook name.  Enter 'rating' show complete list sorted by rating.  Enter 'quit' to end program.  > **quit** |

*Mock-up of program in use. User inputs in* ***bold****.*

##### **Pseudocode**

This pseudocode describes the three main functions of the program (without added validation).

**Function** filterBySubject(list)

subject ← **Input** 'Enter the subject to use as filter: '

**For each** entry **in** list

**If** entry.subject **Contains** subject:

**Display** entry

**End If**

**End For**

**If no matches were found**

print('No entries found with that subject.')

**End If**

**End Function**

**Function** filterByTextbook(list)

textbook ← **Input** 'Enter the textbook name to use as filter: '

**For each** entry **in** list

**If** entry.textbook **Contains** textbook:

**Display** entry

**End If**

**End For**

**If no matches were found**

print('No entries found with that subject.')

**End If**

**End Function**

**Function** sortByRating(list)

textbook ← **Input** 'Enter the textbook name to use as filter: '

*// First, make a copy of the list where ratings are all numbers.*

**Clone** list **to** fixedList

**For each** entry **in** fixedList:

**If** entry.rating **=** 'none':

entry.rating = 0

**End If**

**End For**

*// Next, do a selection sort into a sorted list.*

**Perform selection sort into** sortedList

*// Finally, print the sorted list.*

**Display** sortedList

**End Function**

## Appendix 1: Testing table template

Use the following testing table to check your module's functionality. The completed test table should be included with every submission.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Expected result** | **Actual result** | **Pass/fail** | **Remedy** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Action**: some action, event or input to the system which should provoke a response. Typically, this uses a verb: ‘Click the select button’, ‘Select the third item from the drop down box’, ‘Focus a light on the light sensor’.

**Expected response**: this describes what the system *should* do in response to the action taken. It could start with the words ‘The system shall...’. An example might be ‘display the words “HELLO WORLD” in upper text box’. Be specific and definite.

**Actual response**: this describes what *actually* happened (or didn't happen). Try to gather as much information as possible here. ‘The text “HELLO WOR” was displayed and filled the whole text box’.

**Pass/fail**: if the actual result matches the expected result exactly, the test passes.

**Remedy**: if you can, describe what action needs to be taken to fix the problem. This is not always possible at the time of testing but can be added after the program has been fixed. e.g. ‘The size of the text box needs to be increased to accommodate longer sentences.’

Failed test should be redone. A major change or redesign should cause all tests to be re-run as *regression* tests to ensure that no side-effects of a code change have affected other parts of the program.

## Appendix 2: Example shop data

This table gives an example of the data kept by the second-hand textbook shop. A .CSV file with the same content is optionally provided.

| **Textbook** | **Subject** | **Seller** | **Purchase price** | **Purchaser** | **Sale price** | **Rating** |
| --- | --- | --- | --- | --- | --- | --- |
| Understanding silicon life | Year 10 Aliens | Tina Piercy | 63.20 | Warner Beisner | 75.00 | none |
| First contact: a guide | Year 10 Aliens | Chad Cadieux | 107.40 | Tonja Boughner | 130.00 | none |
| Faster than light propulsion | Year 11 Engines | Scott Rezentes | 103.20 | Genevieve Wiebe | 125.00 | 5 |
| Mapping our quadrant | Year 9 Navigation | Laurena Shattuck | 43.60 | Shawnee Zaremba | 55.00 | none |
| First contact: a guide | Year 10 Aliens | Edythe Spruill | 35.80 | Roseanna Kinnear | 55.00 | none |
| First contact: a guide | Year 10 Aliens | Thelma Colberg | 107.40 | Donny Burkhard | 120.00 | 5 |
| Mapping our quadrant | Year 9 Navigation | Brain Funaro | 21.80 | Ike Caton | 45.00 | none |
| Planets and satellites | Year 9 Planets | Brain Funaro | 47.60 | Shawnee Zaremba | 70.00 | none |
| First contact: a guide | Year 10 Aliens | Randee Ison | 35.80 | Yuko Cirillo | 55.00 | none |
| First contact: a guide | Year 10 Aliens | Chanda Benford | 83.20 | NA | NA | none |
| First contact: a guide | Year 10 Aliens | Jeraldine Salls | 71.60 | Luciana Fey | 95.00 | none |
| Understanding silicon life | Year 10 Aliens | Linda Mulford | 63.20 | Mertie Caraway | 75.00 | 5 |
| Safety on molten worlds | Year 9 Planets | Lilla Edick | 29.80 | Destiny Serrata | 50.00 | 4 |
| Safety on molten worlds | Year 9 Planets | Queenie Oliveros | 59.60 | Camie Giefer | 75.00 | none |
| Mapping our quadrant | Year 9 Navigation | Gertude Olveda | 65.40 | Camie Giefer | 90.00 | none |
| Planets and satellites | Year 9 Planets | Randa Bybee | 71.40 | Linnie Mudd | 85.00 | none |
| Lifeforms and societies | Year 10 Planets | Tina Piercy | 39.60 | Warner Beisner | 50.00 | 3 |
| First contact: a guide | Year 10 Aliens | Olivia Lamm | 107.40 | Tanja Zajac | 130.00 | 5 |
| Understanding silicon life | Year 10 Aliens | Carola Mothershed | 63.20 | Sandie Gierlach | 85.00 | none |
| Safety on molten worlds | Year 9 Planets | Idella Brashears | 119.20 | Glenna Sutcliffe | 130.00 | none |
| Lifeforms and societies | Year 10 Planets | Bev Defalco | 19.80 | NA | NA | none |
| Safety on molten worlds | Year 9 Planets | Arnold Fontanez | 29.80 | Sheron Hakala | 45.00 | none |
| First contact: a guide | Year 10 Aliens | Adria Surprenant | 107.40 | Reyes Whitty | 115.00 | 5 |
| First contact: a guide | Year 10 Aliens | Rob Foust | 107.40 | Epifania Fizer | 125.00 | 4 |
| Understanding silicon life | Year 10 Aliens | Gerry Cowman | 47.40 | Latosha Lindgren | 70.00 | none |
| First contact: a guide | Year 10 Aliens | Lamar Perrone | 143.20 | Ellie Schroth | 155.00 | none |
| First contact: a guide | Year 10 Aliens | Roseanna Shelly | 107.40 | Lorine Shadduck | 120.00 | none |
| First contact: a guide | Year 10 Aliens | Monte Bushnell | 35.80 | Levi Matt | 45.00 | 3 |
| Planets and satellites | Year 9 Planets | Indira Betts | 47.60 | Eliana Youmans | 60.00 | none |
| Lifeforms and societies | Year 10 Planets | Gertie Grimm | 79.20 | Kieth Buettner | 90.00 | 4 |
| Mapping our quadrant | Year 9 Navigation | Dino Maffei | 87.20 | Delmy Michaelsen | 95.00 | none |
| Safety on molten worlds | Year 9 Planets | Dino Maffei | 89.40 | Delmy Michaelsen | 100.00 | none |
| Understanding silicon life | Year 10 Aliens | Edie Rust | 31.60 | Lyn Giuffre | 45.00 | 1 |
| Safety on molten worlds | Year 9 Planets | Gertrudis Coats | 59.60 | Kittie Loiselle | 80.00 | none |
| First contact: a guide | Year 10 Aliens | Man Newell | 71.60 | Remedios Yarborough | 95.00 | none |
| Faster than light propulsion | Year 11 Engines | Lita Logue | 25.80 | Robin Grise | 35.00 | none |
| Mapping our quadrant | Year 9 Navigation | Jacqualine Alfrey | 65.40 | Louella Toppin | 85.00 | none |
| Understanding silicon life | Year 10 Aliens | Joselyn Skates | 63.20 | Edward Dillahunty | 85.00 | none |
| Mapping our quadrant | Year 9 Navigation | Hedwig Marez | 21.80 | Dominga Spagnolo | 40.00 | none |
| Lifeforms and societies | Year 10 Planets | Sacha Brunet | 39.60 | Edward Dillahunty | 55.00 | none |
| First contact: a guide | Year 10 Aliens | Sacha Brunet | 71.60 | Roland Dynes | 85.00 | none |
| Planets and satellites | Year 9 Planets | Christeen Lugar | 23.80 | Liliana Steinert | 35.00 | none |
| Mapping our quadrant | Year 9 Navigation | Xavier Shankle | 87.20 | Rowena Toman | 105.00 | 5 |
| Lifeforms and societies | Year 10 Planets | Lashawna Seman | 79.20 | Chasity Magyar | 105.00 | none |
| Understanding silicon life | Year 10 Aliens | Johnna Roudebush | 31.60 | Marvel Leech | 45.00 | 5 |
| Planets and satellites | Year 9 Planets | Lucille Travis | 71.40 | Malik Liggett | 80.00 | none |
| Planets and satellites | Year 9 Planets | Wendie Urrutia | 47.60 | NA | NA | none |
| Planets and satellites | Year 9 Planets | Fidela Grieve | 71.40 | NA | NA | none |
| Safety on molten worlds | Year 9 Planets | Lajuana Calton | 89.40 | Loyce Leno | 110.00 | none |
| Understanding silicon life | Year 10 Aliens | Reda Cheatham | 31.60 | Layne Temple | 45.00 | none |
| Planets and satellites | Year 9 Planets | Denisha Levar | 23.80 | Madeline Duffer | 40.00 | 2 |
| Mapping our quadrant | Year 9 Navigation | Denisha Levar | 43.60 | Cristina Higgin | 60.00 | none |
| Mapping our quadrant | Year 9 Navigation | Isiah Salvas | 21.80 | Madeline Duffer | 30.00 | 4 |
| Safety on molten worlds | Year 9 Planets | Isiah Salvas | 119.20 | Cristina Higgin | 140.00 | none |
| Faster than light propulsion | Year 11 Engines | Celia Sutterfield | 25.80 | German Mcnamara | 45.00 | none |
| Planets and satellites | Year 9 Planets | Lawanna Dorsey | 95.20 | Carola Garica | 115.00 | none |
| Safety on molten worlds | Year 9 Planets | Anastacia Dones | 119.20 | Georgann Pangle | 130.00 | none |
| Mapping our quadrant | Year 9 Navigation | Stephaine Hamed | 21.80 | Sandie Helwig | 45.00 | none |
| Lifeforms and societies | Year 10 Planets | Lane Bartlow | 39.60 | Doreen Roger | 60.00 | 3 |
| Faster than light propulsion | Year 11 Engines | Colton Rust | 25.80 | Kecia Hanby | 40.00 | none |
| First contact: a guide | Year 10 Aliens | Lane Bartlow | 143.20 | Doreen Roger | 170.00 | none |
| Mapping our quadrant | Year 9 Navigation | Cecily Triano | 21.80 | Tam Bakke | 30.00 | none |
| Lifeforms and societies | Year 10 Planets | Kellee Crivello | 59.40 | Heike Nardi | 80.00 | 5 |
| First contact: a guide | Year 10 Aliens | Sulema Gatlin | 71.60 | Florine Folk | 80.00 | none |
| Safety on molten worlds | Year 9 Planets | Holly Davisson | 29.80 | Cristine Saad | 45.00 | none |
| Understanding silicon life | Year 10 Aliens | Caron Mraz | 47.40 | Twyla Maio | 55.00 | 4 |
| Lifeforms and societies | Year 10 Planets | Caron Mraz | 19.80 | Twyla Maio | 40.00 | none |
| Safety on molten worlds | Year 9 Planets | Dallas Quijano | 29.80 | NA | NA | none |
| Faster than light propulsion | Year 11 Engines | Karen Troxler | 51.60 | Ruthie Calico | 60.00 | none |
| Lifeforms and societies | Year 10 Planets | Willis Calbert | 39.60 | NA | NA | 5 |
| First contact: a guide | Year 10 Aliens | Kenna Holston | 71.60 | Marceline Quach | 90.00 | none |
| Understanding silicon life | Year 10 Aliens | Susy Boswell | 15.80 | Dion Mcalpin | 40.00 | none |
| Mapping our quadrant | Year 9 Navigation | Dian Micek | 21.80 | Elmer Falcone | 45.00 | 3 |
| Understanding silicon life | Year 10 Aliens | Willis Calbert | 15.80 | Janee Grammer | 25.00 | none |
| First contact: a guide | Year 10 Aliens | Terence Defeo | 35.80 | Lorelei Kroon | 55.00 | none |
| Planets and satellites | Year 9 Planets | Myung Krizan | 71.40 | Angeles Ingerson | 80.00 | none |
| Mapping our quadrant | Year 9 Navigation | Alexandria Mcgaugh | 21.80 | Antione Tomes | 35.00 | none |
| First contact: a guide | Year 10 Aliens | Fairy Bourbon | 107.40 | Susann Seale | 115.00 | none |
| Safety on molten worlds | Year 9 Planets | Avelina Rutter | 89.40 | Marvella Amador | 105.00 | none |
| First contact: a guide | Year 10 Aliens | Man Mims | 107.40 | Gino Shugart | 130.00 | none |
| Mapping our quadrant | Year 9 Navigation | Terresa Fikes | 87.20 | NA | NA | none |
| Understanding silicon life | Year 10 Aliens | Liz Shearin | 15.80 | Berta Selden | 40.00 | none |
| Faster than light propulsion | Year 11 Engines | Jade Delcambre | 25.80 | Many Kinsley | 35.00 | 4 |
| Faster than light propulsion | Year 11 Engines | Stephenie Sicard | 103.20 | Kendal Attaway | 130.00 | none |
| Planets and satellites | Year 9 Planets | Esperanza Astin | 95.20 | Jordon Ulibarri | 120.00 | none |
| Safety on molten worlds | Year 9 Planets | Carmela Witmer | 119.20 | Sarita Slocumb | 145.00 | none |
| Lifeforms and societies | Year 10 Planets | Tyisha Eckstein | 19.80 | Lorelei Kroon | 30.00 | none |
| Planets and satellites | Year 9 Planets | Rachael Kilgore | 23.80 | Kayce Witten | 35.00 | none |
| First contact: a guide | Year 10 Aliens | Conrad Kutcher | 71.60 | Lula Whang | 95.00 | 5 |
| Mapping our quadrant | Year 9 Navigation | Esperanza Astin | 43.60 | Jordon Ulibarri | 55.00 | none |
| Lifeforms and societies | Year 10 Planets | Yevette Lara | 19.80 | Latisha Falkowski | 30.00 | none |
| First contact: a guide | Year 10 Aliens | Agnus Mahr | 107.40 | Dane Bohlen | 120.00 | none |
| Faster than light propulsion | Year 11 Engines | Chantel Knepper | 73.20 | NA | NA | none |
| Lifeforms and societies | Year 10 Planets | Keneth Luo | 79.20 | Taina Suriel | 95.00 | none |
| Planets and satellites | Year 9 Planets | Shantell Peevy | 95.20 | Lydia Cutshall | 105.00 | 2 |
| Planets and satellites | Year 9 Planets | Trish Sommer | 71.40 | NA | NA | none |
| Understanding silicon life | Year 10 Aliens | Keneth Luo | 15.80 | Christi Whitfield | 40.00 | none |
| First contact: a guide | Year 10 Aliens | Nelly Perone | 71.60 | Clarence Stoddard | 80.00 | none |
| Understanding silicon life | Year 10 Aliens | Grace Hoffert | 31.60 | Lyndon Edgecomb | 40.00 | none |
| First contact: a guide | Year 10 Aliens | Norberto Fairbank | 71.60 | Joline Perine | 90.00 | none |
| Faster than light propulsion | Year 11 Engines | Marleen Wilkes | 25.80 | Toi Escoto | 50.00 | 5 |
| Faster than light propulsion | Year 11 Engines | Corazon Mccreary | 51.60 | Claude Bonomo | 75.00 | 5 |
| Lifeforms and societies | Year 10 Planets | Thi Eagle | 19.80 | Alphonse Coriell | 45.00 | none |
| Mapping our quadrant | Year 9 Navigation | Renee Cully | 65.40 | NA | NA | none |
| Mapping our quadrant | Year 9 Navigation | Janeth Pera | 87.20 | Theressa Noble | 95.00 | none |
| Safety on molten worlds | Year 9 Planets | Janeth Pera | 89.40 | Antione Tomes | 115.00 | 4 |
| Faster than light propulsion | Year 11 Engines | Celinda Iliff | 51.60 | Rodrigo Domingues | 75.00 | none |