|  |
| --- |
| **Digital Technologies Department**  logo**91076 Coding Assessment (Internal)** |
| Assessment Type: Internal  PRACTICE  Achievement Standard Assessed: 91076 (version 2)  Level: 1  Credits: 3 |

Teacher guidelines

Teachers need to be very familiar with the outcome being assessed by Achievement Standard Technology 91076. The achievement criteria and the explanatory notes contain information, definitions, and requirements that are crucial when interpreting the standard and assessing students.

Setting

Students are required to code a problem on TV viewing data as specified in the student instructions.

Conditions

This is an individual assessment activity. Students will have 1 week of in-class of time to complete it. (*4 lessons of in class time for the practice assessment*)

Resource requirements

Students will require a computer. Software options are Microsoft Word, Excel and Scratch

|  |
| --- |
| **Digital Technologies Department**  logo**91076 Coding Assessment (Internal)** |
| Assessment Type: Internal  PRACTICE  Achievement Standard Assessed: 91076 (version 2)  Level: 1  Credits: 3 |

| Achievement | Achievement with Merit | Achievement with Excellence |
| --- | --- | --- |
| Construct a basic computer program for a specified task | **Skilfully** construct a basic computer program for a specified task. | **Efficiently** construct a basic computer program for a specified task. |

Student instructions

Introduction

Context and issue

A researcher is collecting and analysing data on TV viewing time for various people. Design a program to

determine and print out someone’s name and their average weekly TV viewing time. Data entered is the name followed by the 7 recorded viewing times for the week. Also determine and printout the highest recorded viewing time for that person. For example, if the data entered is:

Wiremu Jones, 3, 1.2, 0, 4, 3.7, 6, 1.4

Output is:

Wiremu Jones

Average viewing time: 2.76 hours

Highest viewing time: 6 hours

You will be assessed on how correct and efficient your program code is, as well as how you have tested and debugged your code for various inputs. Note that you will not be required to plan your solution.

**You are to document your process of testing and debugging.**

Your programming code needs to do these things:

1. Have sequence, selection and iteration control structures
2. Have program code set out clearly and documented with suitable comments
3. Have well-chosen actions, conditions and control structures so that the resulting sequence of actions correctly performs the task and has no unintended behaviour or consequences.
4. Be flexible and robust and have no unnecessary duplication or repetition

This is an individual task. You have 2 hours to complete it. *(4 lessons of in-class time*)

Task 1: Test data for expected inputs

Create a folder in your H drive labelled: Practice AS91076.

Inside this folder create a folder labelled Code.

Open the word document “Record of Test and Debug Practice AS91076” in P:\.......

Save it to your H:\ drive in the Practice AS91076 folder to record your process of testing and debugging.

Put your name and school ID number in the header.

Study the plan given for the solution to the problem.

Draw up a table of test data for expected inputs and calculated outputs on your Record of Test and Debug document.

Task 2: Write code for version #1 of the plan

From the plan for the solution to the problem, write documented code for version #1 of the solution to the problem. Test and debug your code with expected test data, recording the process of testing and debugging.

Task 3: Write code for version #2 of the plan

Write and test your code for version #2 with test data from Task 1. Document what you have tested and debugged.

Task 4: Write code for version #3 of the plan

Write and test your code for version #3 with test data from Task 1. Document what you have tested and debugged.

Ensure that your program is written in a suitable programming language, and:

* includes variables of at least two data types, assignment, predefined actions (eg predefined methods, functions, or procedures), expressions, sequence, selection and iteration control structures
* obtains and uses input from a user, sensors, or other external source.
* the program code is set out clearly and documented with suitable comments
* has been tested and debugged to ensure that it works on a sample of expected inputs.
* that it is correct for expected inputs

UP TO HERE FOR ACHIEVED

Task 5: Create a test data table to cover a range of inputs (some exceptional, or boundary inputs)

Create a **testing table** for exceptional or boundary inputs.

Create version 4 of your code, to handle the input of exceptional or boundary data.

*(Hint: How many hours in a day?)*

Document your process of testing and debugging.

Ensure that :

* you have independently implemented the plan for a basic program in a suitable programming language that uses a procedural structure with well-chosen actions, conditions and control structures
* you have documented the program with variable names and comments that accurately describe code function and behaviour
* you have tested and debugged the program in an organised way to ensure that it works on expected and boundary inputs.

UP TO HERE FOR MERIT

Task 6: Add to the test data table to cover a range of inputs (which may include exceptional, out-of-range, boundary or invalid inputs)

Make additions to the **testing table** for any exceptional, boundary, out-of-range or invalid inputs

Create version 5 of your code, to handle the input of exceptional, boundary, out-of-range or invalid inputs

Document your process of testing and debugging.

Ensure that you have:

* constructed a basic program which uses actions, conditions and control structures effectively to increase the flexibility and robustness of the program
* used an effective procedural structure that results in a well-structured, logical solution to the task
* set out the program code concisely and documented the program with succinct comments that explain and justify decisions
* comprehensively tested and debugged the program in an organised and time-effective way to ensure the program is correct on expected, boundary and invalid inputs.

UP TO HERE FOR EXCELLENCE

Marking Rubric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Student/  Date | Teacher/  date | **Assessment strategies** | **Assessment evidence** |
| **Achieved** |  |  | I have read and understood the problem | Problem statement |
|  |  | I have drawn up a test data table for expected inputs | Test data table |
|  |  | I have coded version 1 of the plan | Program language code– version 1 |
|  |  | I have tested and debugged version 1 with my test data and recorded my testing process | Documented evidence of testing and debugging with test data |
|  |  | I have confirmed that my program so far is correct on expected inputs | Version 1 is correct |
|  |  | I have coded version 2 of the plan | Program language code– version 2 |
|  |  | I have tested and debugged version 2 with my test data and recorded my testing process | Documented evidence of testing and debugging with test data |
|  |  | I have confirmed that my program so far is correct on expected inputs | Version 2 is correct |
|  |  | I have coded version 3 of the plan | Program language code– version 3 |
|  |  | I have tested and debugged version 3 with my test data and recorded my testing process | Documented evidence of testing and debugging with test data |
|  |  | I have confirmed that my program so far is correct on expected inputs | Version 3 is correct |
|  |  | My program code has suitable comments | Program code |
|  |  | My program has well defined variables | Program code |
|  |  | My program has assignment statements | Program code |
|  |  | My program has sequence (steps in order) | Program code |
|  |  | My program has selection (choice of actions) | Program code |
|  |  | My program has iteration (a loop) | Program code |
|  |  | My program will produce a correct result for a set of expected inputs from the user | Documented evidence and confirmation of correct program |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Merit** |  |  | I have added a testing data table for exceptional and boundary inputs | Document of record of work |
|  |  | I have coded version 4 of my program | Program language code– version 4 |
|  |  | My program has well-chosen actions, conditions and control structures | Program code |
|  |  | My program has variable names and comments that accurately describe code function and behaviour | Program code |
|  |  | I have tested and debugged version 4 with my test data in an organised way and recorded my testing process | Documented evidence of testing and debugging with testing data |
|  |  | I have confirmed that my program so far is correct on some exceptional or boundary inputs | Version 4 is correct |
|  |  |  | I have worked independently | Teacher observation |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Excellence** |  |  | I have added to the testing data table for exceptional, boundary, out of range and invalid inputs | Document of record of work |
|  |  | I have coded version 5 of my program | Program language code– version 5 |
|  |  | I have comprehensively tested and debugged version 5 with my test data and recorded my testing process | Documented evidence of testing and debugging with testing data |
|  |  | I have confirmed that my program will check input data for validity; and use constants, variables and derived values in place of literals. | Program language code– version 5 |
|  |  | I have confirmed that my program will correctly handle expected, boundary and invalid inputs; | Version 5 is correct |
|  |  | My program is a well-structured, logical solution to the task and has no unnecessary duplication or repetition | Program language code– version 5 |
|  |  | My program has succinct comments that explain and justify decisions | Program language code– version 5 |
|  |  | My program is flexible and robust | Program language code– version 5 |

Assessment Schedule: Technology 91076 TV research data

NAME:

|  |  |  |
| --- | --- | --- |
| Evidence/Judgements for Achievement | Evidence/Judgements for Achievement with Merit | Evidence/Judgements for Achievement with Excellence |
| The student has provided the following evidence   |  | | --- | | test data table for expected inputs | | documentation of the process of testing and debugging | | implemented a plan for a basic program in a suitable programming language - coding versions 1, 2 and 3 of the given plan | | set out the program code clearly and documented the program with comments | | tested and debugged the program to ensure that it works on a sample of expected inputs. | | code has variables involving at least two types of information (e.g. numeric, characters, text) and assignment | | code has sequence, selection and iteration control structures | | code has predefined actions (e.g. predefined methods, functions, or procedures) | | code has input from a user, sensors, or other external source. | | code produces a correct result for a set of expected inputs from the assessor |   **The student has constructed a basic computer program for a specified task** **of TV viewing times analysis** | The student has met the Achieved requirements and provided the following evidence:   |  | | --- | | added a testing data table for exceptional and boundary inputs | | coded version 4 of the program | | independently implemented the plan for a basic program in a suitable programming language that uses a procedural structure with well-chosen actions, conditions and control structures | | version 4 of program has sequence of actions that correctly perform the task and have no unintended behaviour or consequences. | | documented the program with variable names and comments that accurately describe code function and behaviour | | tested and debugged the program in an organised way to ensure that it works on expected and boundary inputs. | | Code produces a correct result for a set of expected, exceptional and boundary inputs from the assessor |   **The student has skilfully constructed a basic computer program for a specified task** **of TV viewing times analysis** | The student has met the Merit requirements and provided the following evidence:   |  | | --- | | added to the testing data table for exceptional, boundary and invalid inputs | | coded version 5 of the program | | constructed a basic program which uses actions, conditions and control structures effectively to increase the flexibility and robustness of the program | | used an effective procedural structure that results in a well-structured, logical solution to the task with no unnecessary duplication or repetition | | set out the program code concisely and documented the program with succinct comments that explain and justify decisions | | comprehensively tested and debugged the program in an organised and time-effective way to ensure the program is correct on expected, boundary and invalid inputs | | program checks input data for validity; and uses constants, variables and derived values in place of literals. | | Code produces a correct result for a set of expected, exceptional, boundary and invalid inputs from the assessor |   **The student has efficiently constructed a basic computer program for a specified task** **of TV viewing times analysis** |

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the Achievement Standard.

Plan

Context and issue

A researcher is collecting and analysing data on TV viewing time for various people. Design a program to

determine and print out someone’s name and their average weekly TV viewing time. Data entered is the name followed by the 7 recorded viewing times for the week. Also determine and printout the highest recorded viewing time for that person. For example, if the data entered is:

Wiremu Jones, 3, 1.2, 0, 4, 3.7, 6, 1.4

Output is:

Wiremu Jones

Average viewing time: 2.76 hours

Highest viewing time: 6 hours

**Input**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Variable name*** | ***Data type*** | ***How entered*** | ***Purpose of variable*** |
| name | string | From keyboard | Store data entered |
| tvTime | float | From key board in a loop | Store data entered for processing |

**Output**

|  |  |  |  |
| --- | --- | --- | --- |
| ***variableName*** | ***Format*** | ***When it happens*** | ***Details*** |
| name | text | On screen, after data processed for individual |  |
| averageTvTime | Float 2 dp | On screen after all the times have been entered | Suitable comment |
| highestTvTime | Float 1dp | On screen after all the times have been entered | Suitable comment |

Plan for Testing and Debugging

*# version 1*

# to enter the data

Enter name

Loop for 7 times

Enter a TV viewing time

#processing

Display name

#end code

*# version 2*

# to enter the data and

*# determine average viewing time*

Enter name

*Initialise sum to zero*

Loop for 7 times

Enter a TV viewing time

#processing

*Add tvTime to sum*

*Calculate average viewing time*

*Display name and average viewing time*

#end code

*#version 3*

# to enter the data, and

# determine average viewing time, and

*# determine highest viewing time*

Enter name

Initialise sum to zero

*Initialise highestTVTime to zero*

Loop for 7 times

Enter a TV viewing time

#processing

Add tvTime to sum

*Test if tvTime greater than current highestTvTime and code accordingly*

Calculate average viewing time

Display name and average viewing time

*Display highest recorded viewing time*

#end code

*#version 4*

# add in code to handle expected, exceptional and boundary inputs

*#version 5*

# add in code to handle expected, exceptional, boundary and invalid inputs