MCI project Testing plan

Team number: 27 **Project Title:** Integrating Parsons puzzle into MyUni

1. Introduction

1.1. Project Overview

The project is to provide a solution for integrating the Parsons Puzzle into MyUni, that is to design and develop a universal design of parsons puzzle and implement the design into a customised H5P function which will be integrated with the canvas interface.

1.2. Scope of the test plan document

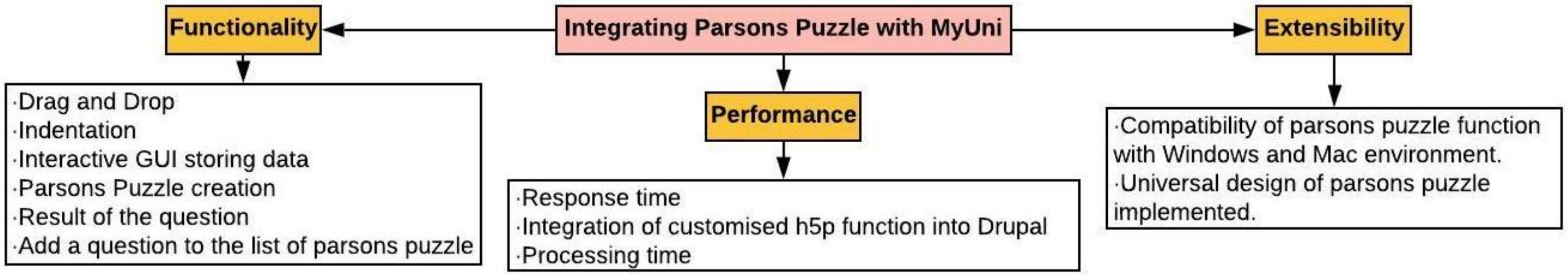
This testing plan gives the strategy to perform the defect, unit and integration testing for the final milestone to fix all the bugs and defects before handing over the deliverable. Besides, some of the actual test cases which have been performed in the first milestone are also included. Moreover, the plan of how the tests should be performed and organised throughout the developmental process is included as well.

2. Test strategy

2.1. Test scope

The purpose of testing is to verify that the function of parsons puzzle type: "Drag and Drop with indentation" works efficiently and fulfils the purpose of its design. There are three modules essential for the working: functionality, performance and extensibility. Unit tests are to be done to every essential function. Validation tests were done to ensure that the system is working properly, especially to check that the functions are operating as intended. Defect tests have been done at every point whenever an error shows up, during developmental process so that parsons puzzle function is compatible with the interface. The outcome of the testing procedure is firstly, the integration of the parsons puzzle into canvas interface and secondly, a testing report.

Figure 1: Features of the project


Integrating Parsons Puzzle with MyUni
Functionality
·Drag and Drop
·Indentation
·Interactive GUI storing data
·Parsons Puzzle creation
·Result of the question
·Add a question to the list of parsons puzzle
Performance
·Response time
·Integration of customised h5p function into Drupal
·Processing time
Extensibility
·Compatibility of parsons puzzle function with Windows and Mac environment.
·Universal design of parsons puzzle implemented.

2.2. Testing report

The testing is divided into three processes: test case creation and tracking, test execution and test reporting. The main functions in the design of the project are drag and drop, fill in the blanks, interactive GUI storing data, parsons puzzle creation and give the result of the question.

Table 1: Testing tool

No.	Process	Tool
1	Test case creation	Microsoft Excel
2	Test execution	**Manual:** Hardware Platform: Acer (15-inch, i7, GPU: Nvidia), Predator (full HD display,15-inch) Software Platform: Acquia Dev Desktop 2, Drupal version 7.69, Node.js, h5p libraries for Drupal. **Automatic:** TestingWhiz Enterprise
3	Test reporting	Microsoft word, PDF

2.3. Test assumptions

- Description is added above the code to explain it.
- The device is interactive and uses scroll bars.
- The user understands the working of parsons puzzle to create a new parsons puzzle.

3. Test execution

3.1. Defect testing

A defect test was done whenever an error occurred. For example, the error when downloading files from GitHub, were caught and solved. There was bug of .git file hidden while cloning it in local computer, and in the properties .git file had to be changed into not hidden. Moreover, checking the working of indentation by dragging the cursor at extreme positions to check any defect.

3.2. Unit testing

After installation of the environment and uploading content on Drupal, Unit tests were done manually to test the following functions: First test was to Add a question to the list of parsons puzzle: Manually adding content of parsons puzzle questions as an instructor, checking whether able to add the content of the quiz in the customised h5p drag and drop function in Drupal. Then, second test was checking Parsons Puzzle creation by publishing the parsons puzzle to undertake the quiz. After that, third test was to verify the “drag and drop” in the parsons puzzle. Furth test was done for testing the ease of working of interactive GUI, and the storage of data. Fifth test was done to check the response time and the performance of the system.

3.3. Integration and release testing

Following the bottom-up approach, the integration testing has been done firstly, by integrating the infrastructure components, then by adding the functional components, which can help in finding errors in the infrastructure integration and the functionality of the parsons puzzle.

4. Testing plan schedule

Overall, testing is in process as the project is developing and plan for testing for the final milestone has been developed. We continue to do testing and debug the code. The unit testing has been done manually for every essential function. The rest of integration and defect testing will be finished before 15/06/2020. The details are presented in the Gantt chart in the Appendix B.

5. Risks

The risks have been identified and impact of the risks control the actions mitigation plan, as to how severely the risk affects the project completion and the urgency to rectify it. Risk causes the event of trigger. The mitigation plan specifies how to tackle and resolve the risk. The details of risk are specified in table 2.

Table 2: Risk impact and mitigation strategy

No.	Risk	Impact	Trigger	Mitigation Plan
1.	Deviation of scope	High	Delay in completing the deliverable of the project	Urgent attention on scope of the project. Focus on scope strictly. Fix problems which deviation might have produced.
2.	Lack of performance of GUI	Medium	Bugs in GUI (buttons not working, complex to use)	Requires attention. Finish once essential functions have been implemented.
3.	Excessive delays	High	Resolving the bugs taking too much time	If the defect doesn’t affect the main functions defined in the scope, it can be fixed later.
4.	Wrong results of testing	Medium	Testers incorrectly analyse boundaries of the partition	Testing as a team. Rechecking the deign of test cases and reviewing of testing by other team mates.

Appendix A

Test description	Input	Expected output	Current output
Add a question to the list of parsons puzzle.	New data that is typed in the text area to create questions.	Storage of the data of questions to be presented once the save button is clicked to create parsons puzzle.	Same as expected.
Test type: Unit Date: 10/05/2020		Responsible: Neha Wali Reviewer: Jialun Han	

Test description	Input	Expected output	Current output
Parsons puzzle creation with a list of questions.	Clicking the save button after adding new data in the text area.	Interactive GUI content with the parsons puzzle created to be solved by user.	Same as expected.
Test type: Unit Date: 10/05/2020		Responsible: Neha Wali Reviewer: Peiting Sun	

Test description	Input	Expected output	Current output
Drag and drop of the set of instructions from the given panel to the solution panel	Movement of cursor (Drag and Drop) by the student of a particular instruction from given panel to solution panel.	Drag and drop of the instruction from the given panel to solution panel smoothly.	Same as expected.
Test type: Unit Date: 10/05/2020		Responsible: Neha Wali Reviewer: Jialun Han	

Test description	Input	Expected output	Current output
Indentation of the instruction in the solution panel.	Movement of cursor from the current position in solution panel to the desired position to indent in solution panel.	Indentation of the selected instruction in the solution panel.	Same as expected.
Test type: Defect Date: 10/05/2020		Responsible: Neha Wali Reviewer: Peiting Sun	

Test description	Input	Expected output	Current output
Adding too many questions(14) in the list of parsons puzzle.	New data that is typed in the text area to create questions.	Storage of the data of questions to be presented once the save button is clicked to create parsons puzzle.	System crashed because of the upper limit of questions being too low. Noted bug and in process of resolving.
Test type: Defect Date: 14/05/2020		Responsible: Jialun Han Reviewer: Neha Wali	

Test description	Input	Expected output	Current output
Choosing no language for the parsons puzzle while adding content for quiz.	Select from a text box a language.	Show error as the language filed is mandatory for the parsons puzzle.	No error shown for choosing no language. Needs to be resolved.
Test type: Defect Date: 14/05/2020		Responsible: Jialun Han Reviewer: Neha Wali	

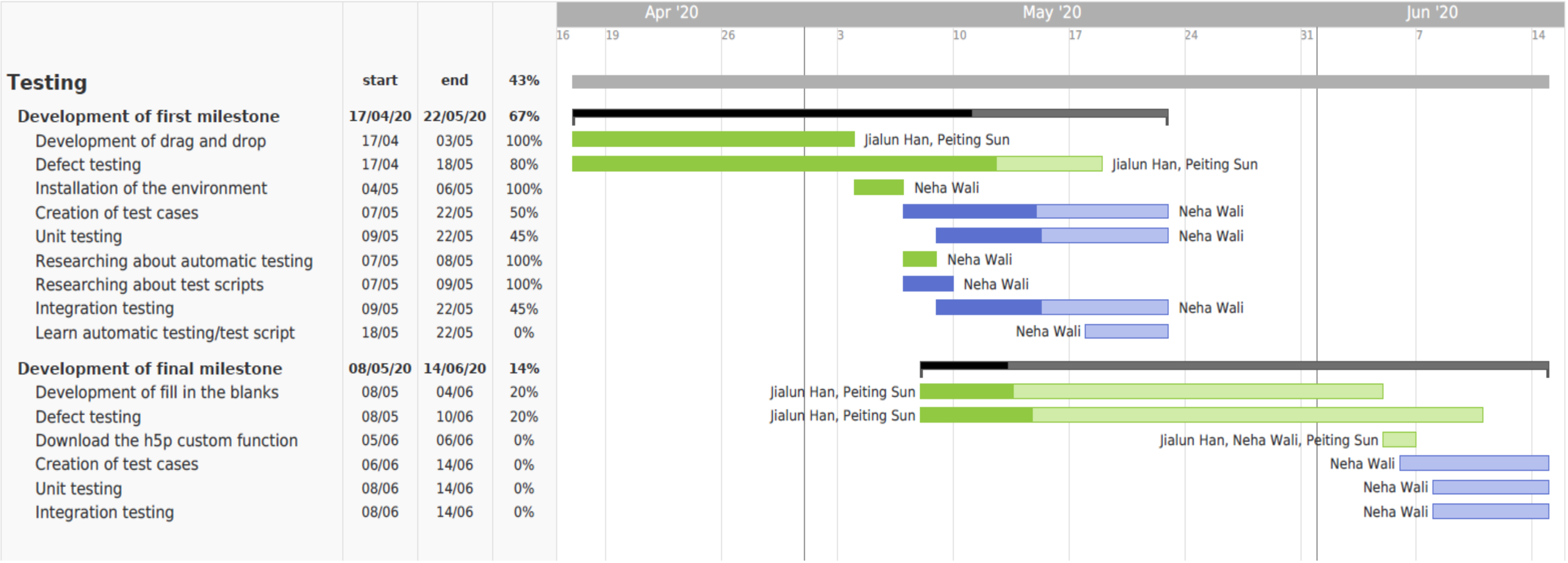
Test description	Input	Expected output	Current output
Compatibility of customised H5P function by running it on Windows 10 and Mac	Downloading H5P function and integrating it with Drupal. Adding content to create quiz.	Creation of quiz.	Same as expected.
Test type: Integration Date: 11/05/2020		Responsible: Neha Wali Reviewer: Jialun Han	

Test description	Input	Expected output	Current output
Checking whether the storage of data of the interactive GUI	Submit the quiz after attempting it, to check the result.	The result is calculated and shown.	Same as expected.
Test type: Integration Date: 14/05/2020		Responsible: Neha Wali Reviewer: Jialun Han	

Test description	Input	Expected output	Current output
The response time of the system to create the quiz and show the result after submission of the student.	New data that is typed in the text area to create questions. Submission of answer by drag and drop of instructions from given panel to solution panel.	Creation of quiz immediately. Showing the result after the student clicks get feedback immediately.	Same as expected.
Test type: Integration Date: 11/05/2020		Responsible: Neha Wali Reviewer: Jialun Han	

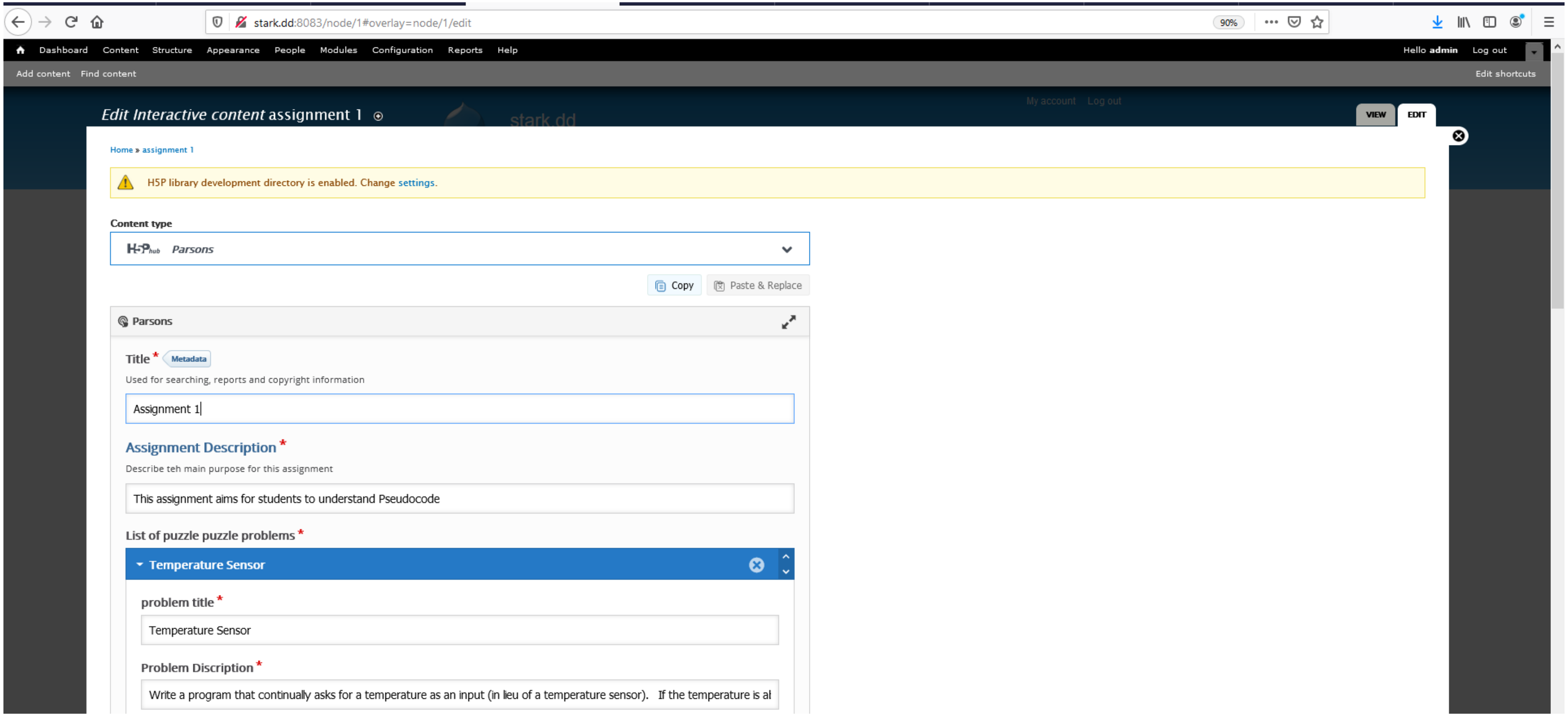
Appendix B

Gantt chart of plan for test schedule



Testing	start	end	43%
Development of first milestone	**17/04/20**	**22/05/20**	**67%**
Development of drag and drop	17/04	03/05	100%
Defect testing	17/04	18/05	80%
Installation of the environment	04/05	06/05	100%
Creation of test cases	07/05	22/05	50%
Unit testing	09/05	22/05	45%
Researching about automatic testing	07/05	08/05	100%
Researching about test scripts	07/05	09/05	100%
Integration testing	09/05	22/05	45%
Learn automatic testing/test script	18/05	22/05	0%
Development of final milestone	**08/05/20**	**14/06/20**	**14%**
Development of fill in the blanks	08/05	04/06	20%
Defect testing	08/05	10/06	20%
Download the h5p custom function	05/06	06/06	0%
Creation of test cases	06/06	14/06	0%
Unit testing	08/06	14/06	0%
Integration testing	08/06	14/06	0%

Appendix C



Edit Interactive content assignment 1

Home » assignment 1

H5P library development directory is enabled. Change settings.

Content type

H5P hub Parsons

Parsons

Title *

Used for searching, reports and copyright information

Assignment 1

Assignment Description *

Describe teh main purpose for this assignment

This assignment aims for students to understand Pseudocode

List of puzzle puzzle problems *

Temperature Sensor

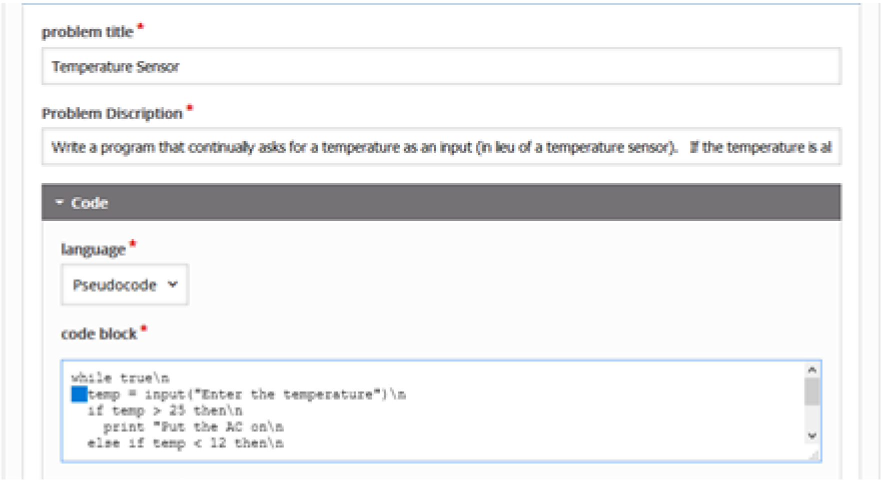
problem title *

Temperature Sensor

Problem Discription *

Write a program that continually asks for a temperature as an input (in lieu of a temperature sensor). If the temperature is al

Figure a: Adding the content of the quiz as an instructor after uploading the customised H5P function into Drupal.



problem title *

Temperature Sensor

Problem Discription *

Write a program that continually asks for a temperature as an input (in lieu of a temperature sensor). If the temperature is al

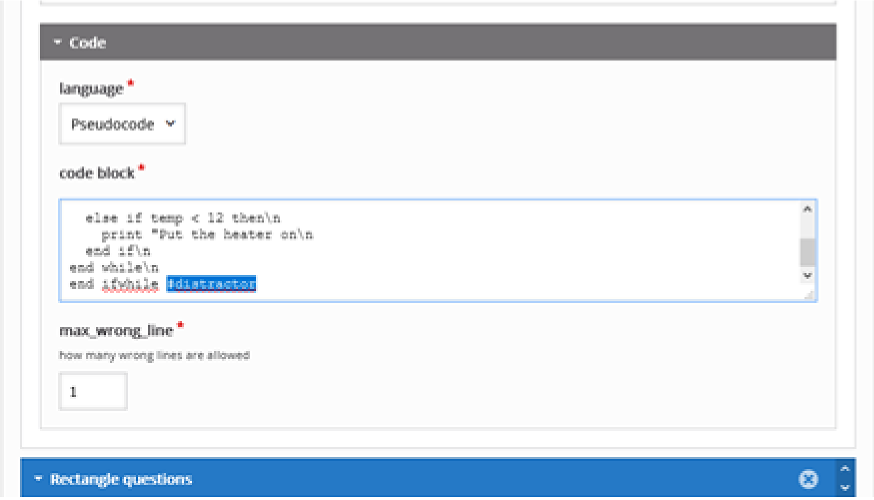
language *

Pseudocode

code block *

```
while true\n
temp = input("Enter the temperature")\n
  if temp > 25 then\n
    print "Put the AC on\n
  else if temp < 12 then\n
```

Figure b: Adding indentation in the code as an instructor.



language *

Pseudocode

code block *

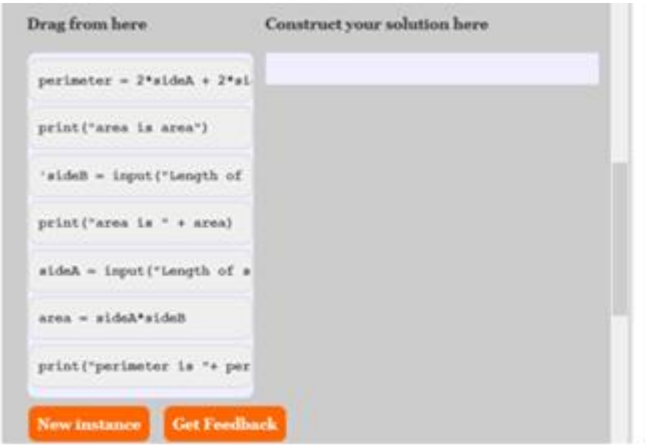
```
  else if temp < 12 then\n
    print "Put the heater on\n
  end if\n
end while\n
end ifwhile #distractor
```

max_wrong_line *

how many wrong lines are allowed

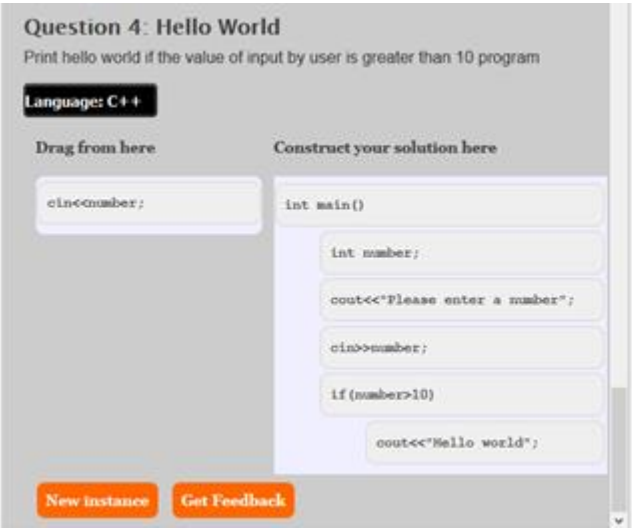
1

Figure c: Adding distractor in the code block as an instructor.



```
perimeter = 2*sideA + 2*si
print("area is area")
'sideB = input("Length of
print("area is " + area)
sideA = input("Length of s
area = sideA*sideB
print("perimeter is "+ per
```

Figure d: Creation of the quiz by the instructor.



Question 4: Hello World

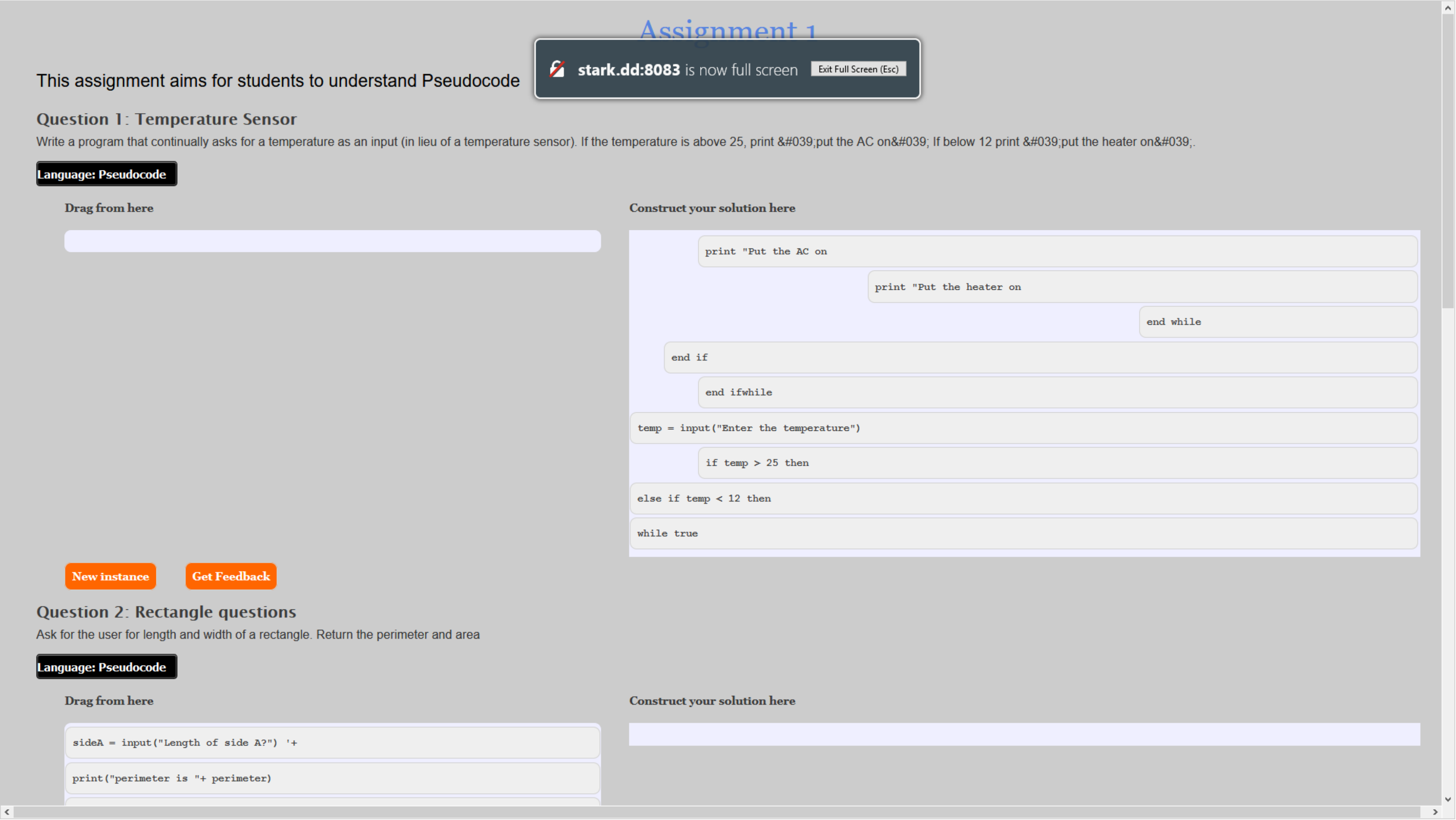
Print hello world if the value of input by user is greater than 10 program

Language: C++

```
cin<<number;
```

```
int main()
    int number;
    cout<<"Please enter a number";
    cin>>number;
    if(number>10)
        cout<<"Hello world";
```

Figure e: Drag and drop and indentation of the instructions as a student.



stark.dd:8083 is now full screen

This assignment aims for students to understand Pseudocode

Question 1: Temperature Sensor

Write a program that continually asks for a temperature as an input (in lieu of a temperature sensor). If the temperature is above 25, print &#039;put the AC on&#039; If below 12 print &#039;put the heater on&#039;.

Language: Pseudocode

Drag from here

Construct your solution here

```
    print "Put the AC on
              print "Put the heater on
                              end while
  end if
    end ifwhile
temp = input("Enter the temperature")
    if temp > 25 then
else if temp < 12 then
while true
```

Question 2: Rectangle questions

Ask for the user for length and width of a rectangle. Return the perimeter and area

Language: Pseudocode

Drag from here

Construct your solution here

```
sideA = input("Length of side A?") '+
print("perimeter is "+ perimeter)
```

Figure f: Resizing the quiz area into full screen as a student.