STOKE DAMEREL
Community College

Cycle 2

Computer Science

Year 11

Name: _____

Tutor: _____

Year 11 Homework Timetable

Monday	English	Ebacc Option A	Option C	
Tuesday	Tassomai	Option B	Option D	
Wednesday	Hegarty	Science	Option C	
Thursday	Ebacc Option A	Tassomai	Option B	Option D
Friday	Hegarty	Science	English	

Tassomai - 50 questions per week
Hegarty - 4 tasks of Hegarty per week

Block A	Block B	Block C	Block D
French	Art	Art	Business Studies
Geography	Business Studies	Business Studies	Catering
History	Child Development	Catering	Dance
Sociology	Catering	Drama	Drama
	Computer Science	History	Geography
	IT	Music	Media Studies
	Media Studies	Photography	Photography
	Sociology	Sport	Sport
	Sport	Travel & Tourism	

Aspire | Achieve | Thrive

GREENSHAW LEARNING TRUST

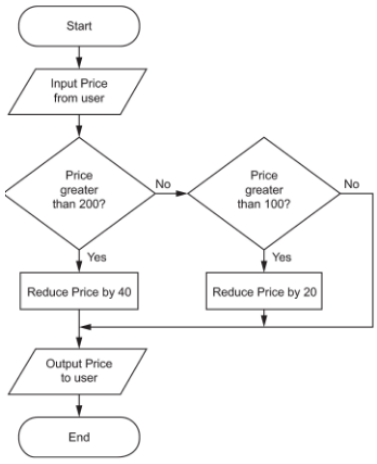
Year 11 - Homework plan Computer Science - Boolean Logic

Week Number	Homework Task	Exam Question
1 **15th November**	**Revision Flash Cards** ● Length Check ● Range Check ● Presence Check	(Q1) SDCC is holding an election with three candidates (1, 2 and 3). An electronic voting booth will be used to allow people to vote. **Write an algorithm that:** -Allows voters to enter either 1, 2 or 3. -Keeps track of how many times each candidate has been voted for. -As soon as one person has finished voting, allow the next person to vote. -At any point allows the official to type in “END”, which will print out the number of votes for each candidate and the total number of votes overall. (6 marks)
2 **22nd November**	**Cornell Notes** ● The purpose of testing. Types of testing: Iterative, Final/terminal	(Q1) A bike dealership uses a computer system to record details of the bikes that it has for sale. Each bike has a make, model, age and number of miles driven. The bike dealership only sells bikes that have fewer than 10 000 miles and are 5 years old or less. Write an algorithm that will: -Ask the user to enter the number of miles and the age of a bike. -Validate the input to check that only sensible values that are in the given range are entered. -Output True if valid data has been entered or False if invalid data has been entered. (5 marks) (Q2) The validation routine from above must be tested with normal, erroneous and boundary test data. Identify suitable test data for each type of test. (3 marks)

	Miles	Age
Normal		
Erroneous		
Boundary		

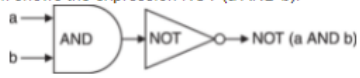
<table>
<tr>
<td>3

29th November</td>
<td>Revision Flash Cards

● Normal Testing
● Boundary Testing
● Invalid/Erroneous Testing</td>
<td>(Q1)
 

Complete the test table for the algorithm above:
<table>
<tr><th>Price input</th><th>Test type</th><th>Expected price output</th></tr>
<tr><td>50</td><td>Normal</td><td></td></tr>
<tr><td>100</td><td>Boundary</td><td></td></tr>
<tr><td>150</td><td>Normal</td><td></td></tr>
<tr><td>200</td><td>Boundary</td><td></td></tr>
<tr><td>250</td><td>Normal</td><td></td></tr>
</table>
(2 marks)</td>
</tr>
<tr>
<td>4

6th December</td>
<td>Cornell Notes

● Simple logic diagrams using the operators AND, OR and NOT</td>
<td>(Q1)
The following logic diagram shows the expression NOT (a AND b).
 

Complete the missing boxes in the truth table below to show the value of NOT (a AND b) output for each possible set of values of a and b.
<table>
<tr><th>a</th><th>b</th><th>NOT (a AND b)</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td></td><td>1</td></tr>
<tr><td>1</td><td>0</td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
(4 marks)</td>
</tr>
<tr>
<td>5

13th December</td>
<td>Revision Flash Cards

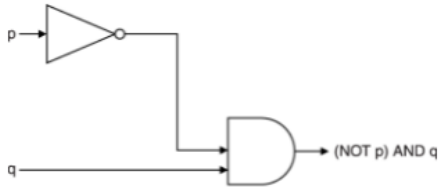
● Combining Boolean operators using AND, OR and NOT</td>
<td>(Q1)
Fig. 1 is a circuit diagram.
 

Fig. 1
Complete the truth table for Fig. 1.
<table>
<tr><th>p</th><th>q</th><th>(NOT p) AND q</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
(3 marks)</td>
</tr>
</table>

6 **3rd January**	**Cornell Notes** ● The characteristics of a compiler and an interpreter	(Q1) Daniel is planning to create a computer game using a high-level programming language. Daniel can use either a compiler or an interpreter to translate the code. Describe two differences between how a compiler and an interpreter would translate Daniel's computer game. (4 marks)
7 and 8 **10th January** **17th January**	**Cornell Notes / Mind Map / Revision Flash Cards** ● All Topics	
9 **24th January**	**Plug the gaps** ● All gaps in knowledge from Assessment.	
Week Number	**Homework Task**	**Exam Question**

Week 1: Defensive design considerations:

Keywords

Authentication - Checking the values that have been entered against existing ones.

Commenting - The process of adding notes to code to explain what that section will do.

Naming Convention - The appropriateness of the names given to functions, variable and iteration statements.

Knowledge

- Defensive design is the process of making sure that a program can become more robust and less prone to errors.
- Validation is a method of creating a robust program because it makes sure that data can only be entered within a particular format (i.e. Number form or date format).
- This will work alongside verification as this will check that the data entered has been entered correctly on more than one occasion.
- Types of defensive design include, but not limited to:

Check Types:	Definition:
Range check	A number or date is within a sensible/allowed range
Type check	Data is of the right type, such as integer, letter or text
Length check	Text entered is not too long or too short – for example, a password is between 8 and 15 characters
Presence check	Checks that data has been entered, i.e. the field has not been left blank
Format check	Checks that the format of, for example, a postcode or email address is correct

Week 2: The purpose of testing. Types of testing: Iterative, Final/terminal:

Keywords

Iterative - Rerun a test after a change to check the outcome.

Final/Terminal - Once a program has all of the modules completed, the program is run as a whole.

Test Data - The actual values that will be entered into a test.

Expected Outcome - This is a prediction before the test has been carried out.

Syntax Error - Spelling/Grammar mistake.

Logic Error - Program runs but not as expected. I.e. Incorrect operator used.

Knowledge

- The purpose of testing is to check for different types of errors.
  - Errors that can be checked are Logic and Syntax errors.
- To determine whether a program is robust and up to standard, there is the use of a test table:

```
answer = int(input("What is 2 * 10? "))
while answer != 20:
    print("Incorrect")
    answer = int(input("What is 2 * 10? "))
print("It's about time.")
```

Test Number:	Test Description:	Test Data:	Expected Outcome:	Actual Outcome:	Comment:	Screenshot Evidence:
1	Can the user enter a float number?	3.6	The program will produce an error message because it looks for integer values.	The program did produce an error message because of the int(input()).	Test Successful	See picture 1
2	Can the user enter a string value?	"Hi"	The program will allow the value to be entered and displayed to the screen.	The program produced an error message because it is looking for integer values.	Test Unsuccessful. INT was included in the input.	

Week 3: Selecting and using suitable test data: Normal, Boundary, Invalid/Erroneous:

Keywords

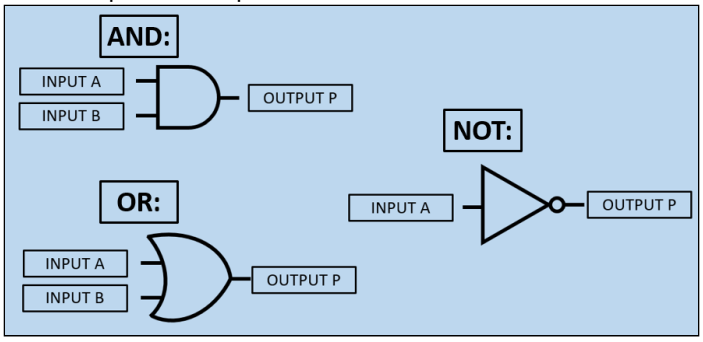
Normal - Values that are to be expected.

Boundary - Entering the smallest and largest value that the program should accept without crashing.

Invalid/Erroneous - Entering a value that is not expected. I.e. Incorrect data type or a value exciting the specified range.

Knowledge

- When selecting test data, it is important to consider the structure of that data.
- Each test will require specific values to be entered. This could come down to:
  - Length
  - Data Type:
    - Float - Decimal
    - Integer - Whole Number
    - String - All characters are accepted
    - Char - One single letter
    - Boolean - True or False
- Using test tables, it is important to record expected and actual results to show how a test should have looked compared to what actually happened.
- An example test would be to enter a string value ("a") when entering an integer for an **Age** variable.
  - **Expected** - The program will reject the input because the value is the wrong data type.
  - **Actual** - The program did indeed reject the input because it was looking for an **integer** value but was provided a **string** value instead.

Week 4: Simple logic diagrams using the operators AND, OR and NOT: (You should also recap Week 1 this week):	
Keywords	**Knowledge**
AND - Both inputs will need to have a TRUE value. **OR** - Only ONE input will need to have a TRUE value. **NOT** - This will invert the input value.	• Logic gates are used to make decisions so that electrical outputs only 'turn on' when the correct logic sequence has been applied. Each logic gate has a name that helps to describe how different inputs will determine the possible outputs. 

Week 5: Combining Boolean operators using AND, OR and NOT: (You should also recap Week 2 this week)	
Keywords	**Knowledge**
Truth Table - Used to determine the outcome of 2 inputs. **AND Notation** - ^ **OR Notation** - V **NOT Notation** - ¬	• A truth table is a simple grid that shows each input and every possible output. We can use these to understand what our circuit is doing. This is especially useful when we start combining logic gates. • We can also express our logic gates using Boolean algebra, where we write down the logic of our gate as a Boolean expression.

AND:

A	B	P
0	0	0
0	1	0
1	0	0
1	1	1

OR:

A	B	P
0	0	0
0	1	1
1	0	1
1	1	1

NOT:

A	P
0	1
1	0

Week 6: The characteristics of a compiler and an interpreter (You should also recap Week 3 this week)	
Keywords	**Knowledge**
Assembly language - A low level language used to represent instructions for the computer. **Assemblers** - Converts assembly language programs into machine code. **Compiler** - This will check for any syntax errors within a program. They will be highlighted in a log. **Editor** - The area where code and programs can be written.	• Programs written in some high level languages are 'compiled' to get them into object code, which the processor can then use. A compiler is a type of translator, which takes an entire program after it is finished (the source code) and converts it into object code in one complete go. Unlike assemblers, a single keyword in a compiled program will get converted into many machine code instructions. If there are any problems with the code, the compiler will report these problems at the end of the compilation process. If the programmer makes any corrections, then the whole program has to be re-compiled again. Once compiled with no errors, the object code can then be run by the processor. • The key difference between a compiler and interpreter here is that the first line of the source code is translated and then run by the processor, and then the second line is translated and run by the processor, and then the third and so on, until the program has finished. Unlike assemblers, a single keyword in an interpreted program will get converted into many machine code instructions.

Week 7 and 8: Preparing for Assessment
Self-quiz the knowledge covered in Weeks 1 - 6

WEEK 1

Date……

SDCC is holding an election with three candidates (1, 2 and 3). An electronic voting booth will be used to allow people to vote.

Write an algorithm that:

-Allows voters to enter either 1, 2 or 3.

-Keeps track of how many times each candidate has been voted for.

-As soon as one person has finished voting, allow the next person to vote.

-At any point allows the official to type in “END”, which will print out the number of votes for each candidate and the total number of votes overall.

(6 marks)

Answer:-

WEEK 2

Date……

(Q1) A bike dealership uses a computer system to record details of the bikes that it has for sale. Each bike has a make, model, age and number of miles driven. The bike dealership only sells bikes that have fewer than 10 000 miles and are 5 years old or less.

Write an algorithm that will:

-Ask the user to enter the number of miles and the age of a bike.

-Validate the input to check that only sensible values that are in the given range are entered.

-Output True if valid data has been entered or False if invalid data has been entered.

(5 marks)

Answer:-

(Q2) The validation routine from above must be tested with normal, erroneous and boundary test data.
Identify suitable test data for each type of test.

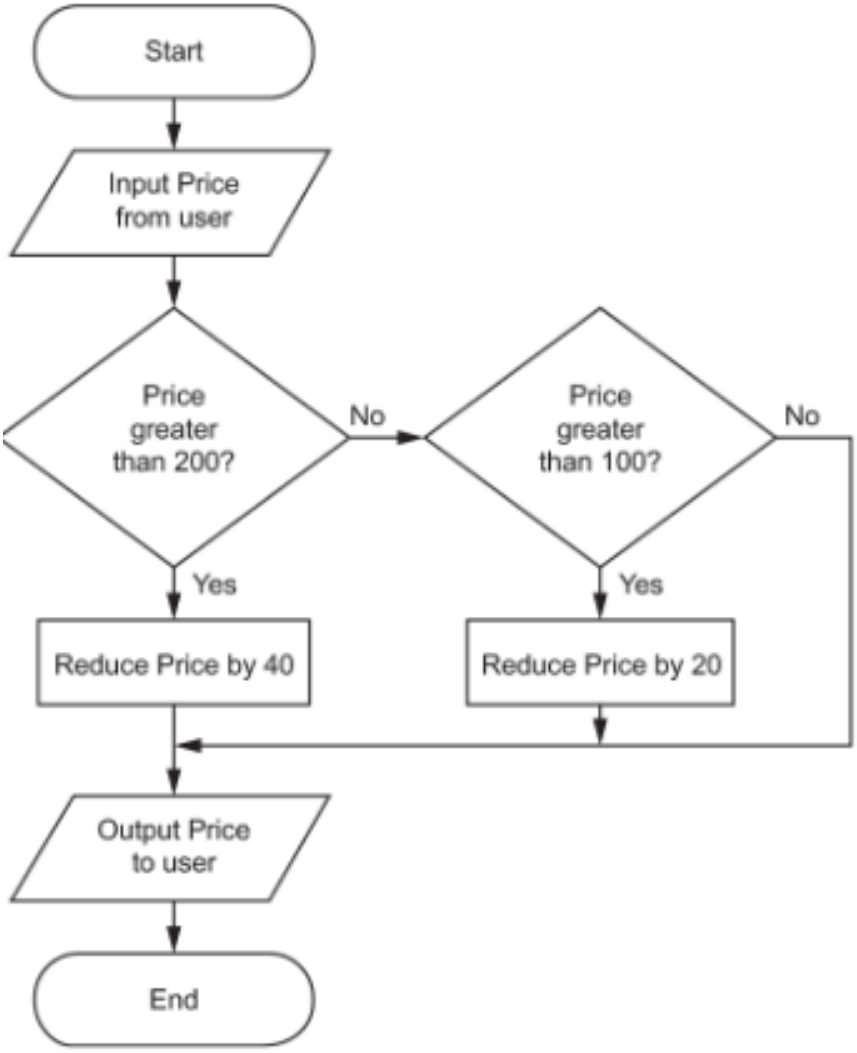
(3 marks)

Answer:-

	Miles:	**Age:**
Normal:		
Erroneous:		
Extreme:		

WEEK 3

Date……



Complete the test table for the algorithm above:

(2 marks)

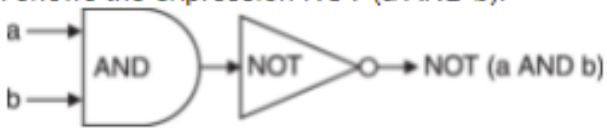
Answer:-

Price Input:	Test Type:	Expected price output:
50	Normal	
100	Boundary	
150	Normal	
200	Boundary	
250	Normal	

WEEK 4

Date……

The following logic diagram shows the expression NOT (a AND b).



Complete the missing boxes in the truth table below to show the value NOT (a AND b) that will be output for each possible set of values of a and b.

(4 marks)

Answer:-

a	b	NOT (a AND b)
0	0	1
0		1
1	0	

WEEK 5

Date……

Fig. 1 is a circuit diagram.

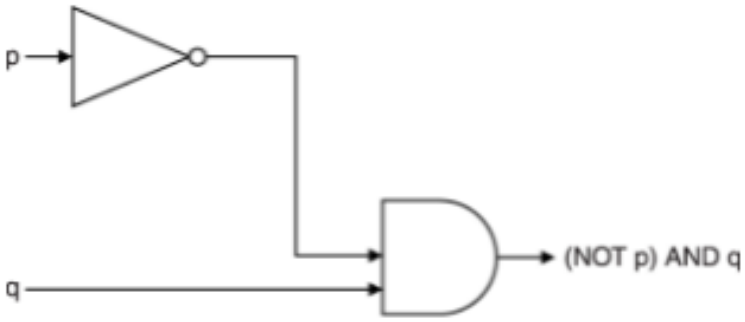


Fig. 1

Complete the truth table for Fig. 1.

(3 marks)

Answer:-

p	q	(NOT p) AND q
0	0	0
1	0	0

WEEK 6

Date……

Daniel is planning to create a computer game using a high-level programming language.
Daniel can use either a compiler or an interpreter to translate the code.
Describe two differences between how a compiler and an interpreter would translate Daniel's computer game.

(4 marks)

Answer:-

STEP 2: CREATE CUES

What: Reduce your notes to just the essentials.

What: Immediately after class, discussion, or reading session.

How:

- **Jot down key ideas, important words and phrases**
- **Create questions that might appear on an exam**
- **Reducing your notes to the most important ideas and concepts improves recall. Creating questions that may appear on an exam gets you thinking about how the information might be applied and improves your performance on the exam.**

Why: Spend at least ten minutes every week reviewing all of your previous notes. Reflect on the material and ask yourself questions based on what you've recorded in the Cue area. Cover the note-taking area with a piece of paper. Can you answer them?

STEP 1: RECORD YOUR NOTES

What: Record all keywords, ideas, important dates, people, places, diagrams and formulas from the lesson. Create a new page for each topic discussed.

When: During class lecture, discussion, or reading session.

How:

- **Use bullet points, abbreviated phrases, and pictures**
- **Avoid full sentences and paragraphs**
- **Leave space between points to add more information later**

Why: Important ideas must be recorded in a way that is meaningful to you.

STEP 3: SUMMARISE & REVIEW

What: Summarise the main ideas from the lesson.
What: At the end of the class lecture, discussion, or reading session.
How: In complete sentences, write down the conclusions that can be made from the information in your notes.
Why: Summarising the information after it's learned improves long-term retention.

Date / / **Topic** **WEEK 2**

Questions	Notes

Summary

Date / / **Topic**

Questions	Notes

Summary

Date / / **Topic**

WEEK 4

Questions	Notes

Summary

Date / / **Topic**

Questions	Notes

Summary

Date / / **Topic** **WEEK 6**

Questions	Notes

Summary

Date / / **Topic**

Questions	Notes

Summary

Revision Page

Revision Page

Revision Page

Revision Page

Revision Page

Revision Page

Revision Page

Revision Card on Defensive design considerations: Anticipating misuse, Authentication 1. Why would we use a length check? 2. Describe what a presence check is. 3. How would a format check become beneficial? 4. What does the term ‘validation’ mean? 5. Provide 2 examples of 2-factor authentication. 6. Why should we use naming conventions in programming?	**Answers**

Revision Card on Selecting and using suitable test data: Normal, Boundary, Invalid/Erroneous: 1. What does Erroneous Testing mean? 2. How does Boundary Testing work? 3. Provide an example of when Normal testing would be used. 4. What is a Test Table used for? 5. Why should we screenshot our tests? 6. What is the purpose of using an Expected Results column in a Test Table?	**Answers**

Revision Card on Combining Boolean operators using AND, OR and NOT: 1. Why do we use a Truth Table? 2. 1 AND 0 will return what output? 3. 1 OR 1 will return what output? 4. What does NOT do to an input? 5. What does ^ represent? 6. What does the ¬ symbol represent?	**Answers**