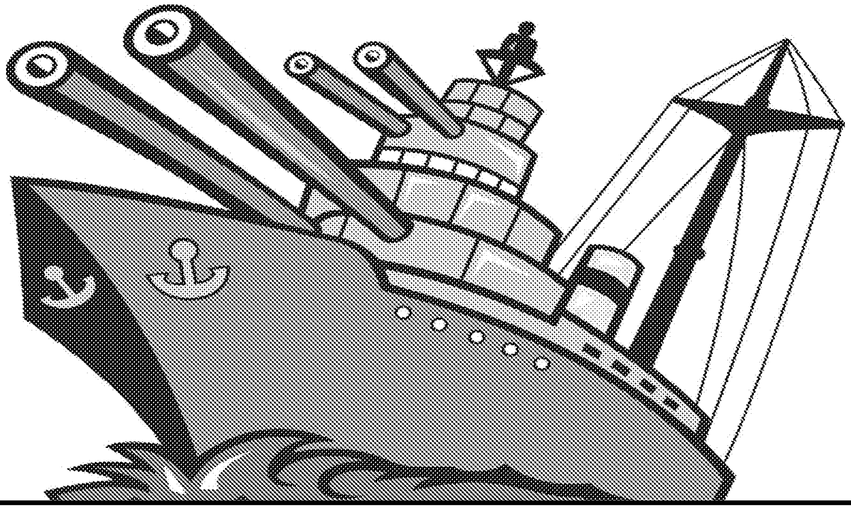
Zig Zag Education

Computer Science

AS | AQA | 7516

2015 specification
for the 2016 AS exam

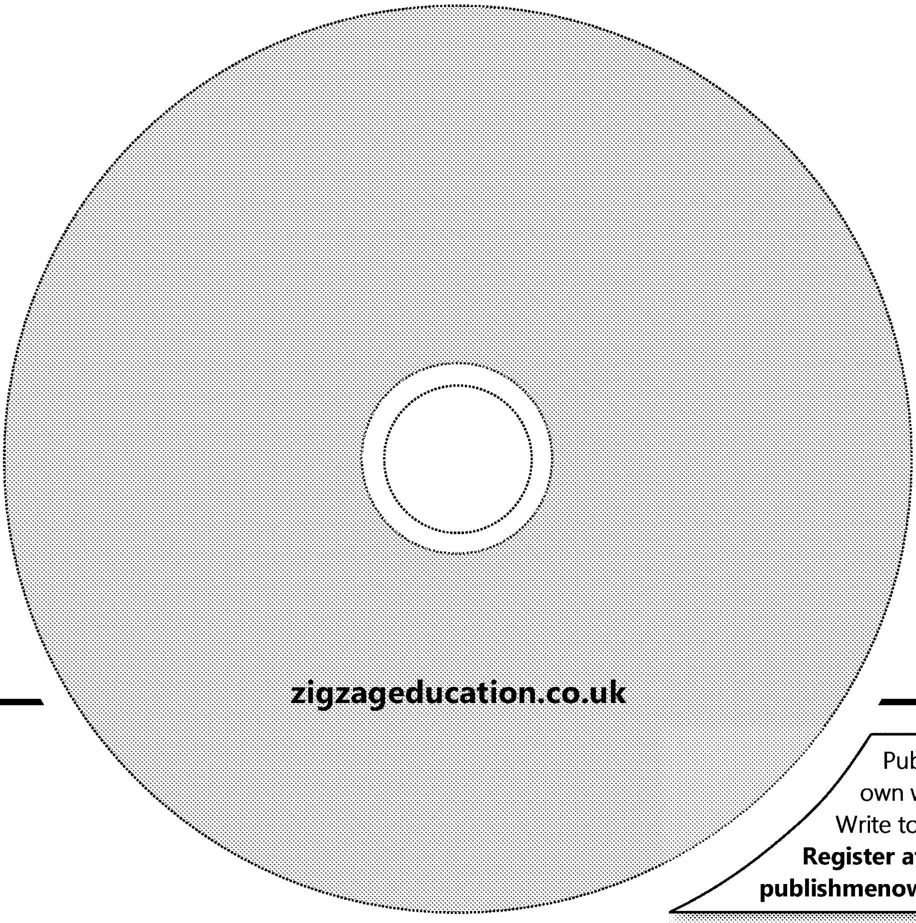


PAPER 1 EXAM RESOURCE PACK 2016

AQA WARSHIPS

for AS AQA Computer Science

VB .NET



AS2/ 6500

POD 6504

zigzageducation.co.uk

Contents

Teacher's Introduction

This pack is designed to help you support your students taking the AQA Computing Paper 1 examination.
It is based on the AQA Paper 1 'AQA Warships' preliminary material (VB .NET) – for examination June 2016.

① **Pre-release Commentary** (for teachers)
A detailed overview of the skeleton program, describing all VB .NET code elements and routines.

This section is designed to help you get to grips with the program, so that you can feel confident helping your students. This commentary is not designed to be given to students before they have explored the code for themselves, and if used in this way could lead to misconceptions of how the program works.

② **Structure Chart Activity**
A partially incomplete diagram for students to complete while getting to grips with the skeleton program. Any missing routines and variables must be added to the diagram. A completed version is provided in the solutions section at the back of the resource.

③ **Programming Theory Questions**
Theory questions test students' understanding of the 'AQA Warships' code, like Section B in the Paper 1 exam. These are provided in both write-on and non-write-on format.

④ **Programming Exercises**
Modification exercises put students' programming skills to the test, like Section C in the Paper 1 exam. An Electronic Answer Document (EAD) and the modified VB .NET code are provided on the CD.

Answers and solutions for the structure chart activity, theory questions and programming exercises are provided from page 21 onwards. Note that for the programming exercises in particular, these are example solutions and you must use your discretion to award marks accordingly where there are valid alternative solutions.

The **Appendices** contains some additional resources, including:

- Further modifications worksheet: a template for brainstorming further enhancements to the skeleton program. This is suggested as a group activity, so that students (and the teacher) can share their ideas, thus increasing the likelihood of covering every area that will come up in the exam.
- Electronic Answer Document (EAD) printout: hard copy version of the file on CD (for reference).

The accompanying CD includes the following files (inside the VB folder):

- **MODIFIED_VB_CODE.txt** – text file containing the additional and/or modified program code as shown in the mark scheme for section ④ (from page 24).
- **PAPER1_EAD.docx** – Electronic Answer Document for completing sections ③ and ④

This resource is intended to supplement your teaching only. It is the teacher's responsibility to decide how to use this resource to assist themselves and their students appropriately. You may simply wish to read this material to better inform yourself and to help you prepare your lessons and to give you ideas for your teaching. You may also consider whether it is appropriate to hand out some of the sheets for reference and to use some of the activities for classwork or homework. You may also consider whether it is appropriate to hand out the booklet to be worked through by your students more independently. As with all pre-release material, it is the teacher's responsibility to decide in what way to assist their students, and to decide how this resource in particular can be used to fit into that assistance.

The resources here are provided as an interpretation of the pre-release material. The author does not have any special knowledge of what to expect on any particular exam.

Programming Exercises: Teacher

Suggested Question Combinations

It is not envisaged that a student would complete all questions in a 1-hour perio
One approach is to get students to work through all the questions under 'open-b
be followed up by setting combinations of the questions under test conditions s

- No access to previously created code
- No access to notes
- No access to the Internet
- No collaboration
- Strict time limit

Suggested question combinations and time limits for these tests are as follows:

Q1, Q2 & Q4	25 minutes
Q3, Q5, Q6 & Q7	30 minutes
Q8 & Q9	20 minutes
Q10 & Q11	25 minutes

Q8 & Q12	30 minutes
Q13 & Q15	60 minutes
Q8 & Q14	35 minutes

It is also useful (and fun) to get students to come out and solve a question 'live' a
classmates.

Possible Additional Questions

1. When the game has finished, tell the user how accurate they were as a perce hits by the total number of shots. E.g. 10 hits, 30 shots = 33% hit rate. Only d
2. One shot sinks a ship.
3. Sea mine is placed on the board. If the player hits it, they lose and the game
4. Change the game so the fleet is five Battleships.
5. Create a two-player game.
6. Change the blast radius so that a torpedo also hits ships in adjacent squares
7. Change the dimensions of the board.
8. Create the option to send a sonar ping down a column or row which tempo ships.
9. Add an ammo store to the board. If the player hits it they get 10 more torpe
10. Change the program so that both coordinates are entered as one input.
11. Make each ship type have a default orientation.
12. Ask for the user's name at the start of the game, and when they win show th [name]!"
13. Allow user to go back to the main menu
14. Change the torpedo to a missile that obliterates a 9 square block.
15. Change the game so that the user places the ships and the computer fires th
16. Adapt the missile task (above) so that the user can choose whether to use a fire a maximum of 2 missiles
17. Add a main menu option which will allow you to select which ships are to be
18. Enhance the computer player in task 15 further so that if it hits a square it w squares until a ship is sunk

AQA WARSHIPS

Description of the Program

The program is designed to play a game which is similar to Battleships.

There are five ships hidden on a 10-by-10 board. The players takes shots at diffe column (0—9) and a row (0

The ships and sizes are as follows:

- Aircraft Carrier — 5 cells
- Battleship — 4 cells
- Submarine — 3 cells
- Destroyer — 3 cells
- Patrol Boat — 2 cells

Ships can be either horizontal or vertical on the board.

The program consists of one constant (TrainingGame) which holds the filename the board. This is then populated into Board (a two-dimensional array of Chars) cell are: — (empty sea), A (a piece of aircraft carrier), B (a piece of battleship), S (of destroyer), P (a piece of Patrol Boat), m (an empty square that has already bee contained a piece of ship and has been hit).

The program has two possible starts: the first is where the position of the ships i second where random positions for the ships are generated by the computer. T additional code as the ships cannot overlap or go off the board and this is check

The game proceeds by asking the player for a column and then a row. The prog at this index in the Board array. If it is a — this is then replaced by an m. If it is a this is replaced by an h. If this position already contains an m or an h, a message fired here is displayed.

If a position off the board is entered, the program will stop functioning.

To complete and end the game you must sink all parts of each ship. There is no a player may take. The player can keep firing until they have hit every square.

Description of Program Elements

The program consists of several routines to determine the validity of moves and who has won.
The program elements that are used are described in order below.

Element	Type	Description
TShip	User-defined data type which contains the data name and size	Stores the name and size of a ship
Ships	An array of TShip	Stores the name and size of all the shi
Board	A two-dimensional array of characters	Stores the current state of the board
TrainingGame	A string constant	Stores the filename of the training file
MenuOption	An integer variable	Used to store what number the user ha
Row	An integer variable	Used to store the row on the board
Column	An integer variable	Used to store the column on the board
Orientation	A char variable	Stores direction of a ship: V for vertica
HorV	An integer variable	Used to randomly generate the orienta horizontal

Description of Program Routines

The program functions Ⓕ and procedures Ⓟ are described below.

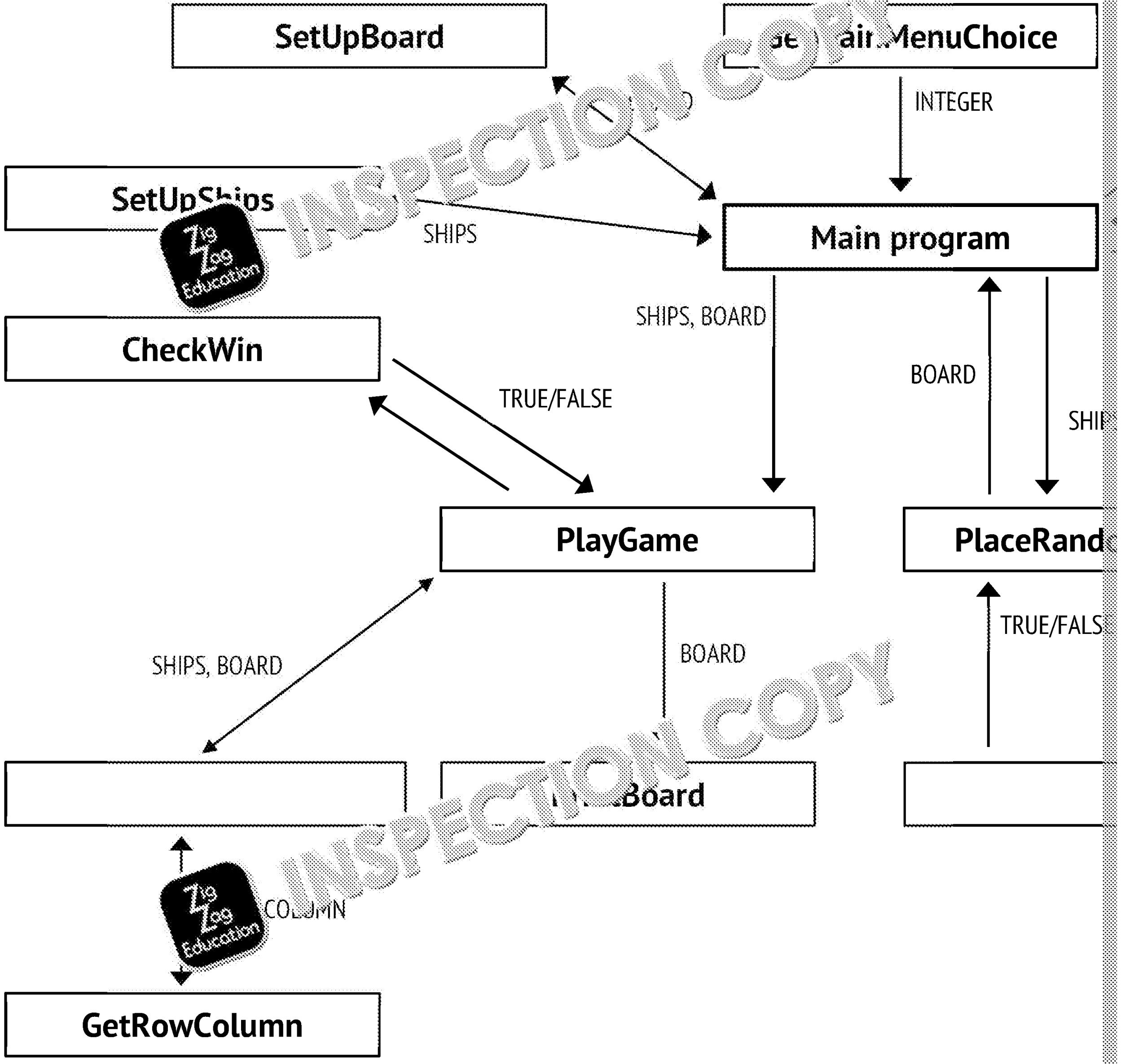
Routine	Description	
CheckWin Ⓕ	Receives: Board Returns: Boolean Called from: PlayGame	Checks every position in board to Returns false if it finds a piece Returns true if it checks every pos
DisplayMenu Ⓟ	Receives: nothing Returns: nothing Called from: Main	A simple procedure that prints op
GetMainMenuChoice Ⓕ	Receives: nothing Returns: integer Called from: Main	Handles the user's menu choice: 1. Prompts the user to enter 2. Returns that number
GetRowColumn Ⓕ	Receives: nothing Returns: integer array Called from: MakePlayerMove	1. Prompts the user for a co 2. Changes the value of the 3. Prompts the user for a ro 4. Changes the value of the
LoadGame Ⓟ	Receives: Filename, Board Returns: nothing Called from: Main	1. Reads the data contained 2. Uses a variable called Lin 3. Then chops Line into ind 4. Repeats for all 10 rows 5. Closes the file
MakePlayerMove Ⓟ	Receives: Board, Ships Returns: nothing Called from: PlayGame	1. Receives the row and col 2. Checks whether that posi 3. Checks whether that posi 4. If neither 2 nor 3 are true

Routine	Description	
PlaceRandomShips ⓟ	Receives: Board, Ships Returns: nothing Called from: Main	This procedure is n It generates a rand ship runs horizonta It then uses the fu running through th doesn't run off the If not, another pos been placed.
PlaceShip ⓟ	Receives: Board, Ship, Row, Column, Orientation Returns: nothing Called from: PlaceRandomShips	Places the ships o Uses For loop that Ship.size). The loo vertical ship (so th placing a horizonta The board is popul ship.
PlayGame ⓟ	Receives: Board, Ships Returns: nothing Called from: Main	Starts a game and 1. Sets the Boole 2. Starts a condit while it is false 2.1. Displays 2.2. Gets the 2.3. Checks to GameWo

Routine	Description	
PrintBoard Ⓟ	Receives: Board Returns: nothing Called from: PlayGame	Displays the board: 1. Starts off by displaying a mes 2. Next, For loop is used to pri 3. Nested For loops now display 3.1. Prints the row number 3.2. Second For loop works it 3.2.1. An empty square is 3.2.2. A square with ship 3.2.3. Anything else (a hit 3.2.4. A separator is displa
SetUpBoard	Receives: Board Returns: nothing Called from: Main	1. Cycles through all positions o 1.1. Assigns all positions on t Some of these dashes will be repl
SetUpShips Ⓟ	Receives: Ships Returns: nothing Called from: Main	Initialises the ships in the array (u Sets the name of each ship Sets the size of each ship
ValidateBoatPosition Ⓕ	Receives: Board, Ship, Row, Column, Orientation Returns: Boolean Called from: PlaceRandomShips	Checks to see whether it is possib Does the boat run off the edge of 1. If the row number plus the sh go off the edge of the board. 2. If the column number plus th it will go off the edge of the b 3. If the ship is vertical: 3.1. A For loop scans along t 3.1.1. If a position isn't em 4. If the ship is horizontal: 4.1. A For loop scans along t 4.1.1. If a position isn't em 5. If this part of the function is r returned.

Routine	Description
Main Ⓟ	1. Declares (creates) an empty two-dimensional array of chars to store 2. Declares (creates) an empty array of TShips to store the fleet details 3. Declares a variable to store what menu option has been selected an 9) 4. Starts a conditional loop that continues until the user selects option 4.1. Populates the board with data by calling SetUpBoard (this would re 4.2. Populates the ships with data by calling SetUpShips 4.3. Displays the menu by calling DisplayMenu 4.4. Calls GetMainMenuChoice to get the user's choice and stores it 4.5. If the user picks option 1: 4.5.1. The board is populated by the ships in random locations 4.5.2. The game is started 4.6. If the user picks option 2: 4.6.1. The board is populated from the training text file 4.6.2. The game is started

AQA WARSH


SetUpBoard
Ge ai.MenuChoice
INTEGER
SetUpShips
SHIPS
Main program
SHIPS, BOARD
BOARD
SHIP
CheckWin
TRUE/FALSE
PlayGame
PlaceRand
TRUE/FALS
SHIPS, BOARD
BOARD
Board
COLUMN
GetRowColumn

Programming Theory Questio

These questions refer to the Preliminary Material and require you to loa
but do not require any additional programming.

1. State the name of an identifier for:

(a) An array or list variable

....

(b) A subroutine that has five parameters

....

(c) A variable that is used to store a whole number

....

(d) A subroutine that returns one or more values

....

(e) A variable that stores a Boolean value

....

2. Look at the function ValidateBoatPosition.

What is the purpose of the variable Orientation?

....

....

....

3 What data is stored for each ship?

....

....

....

4. Look at the procedure [illegible].

What is the purpose of the Do Until loop?

....

....

....

....

5. Give an example of a declaration and assignment statement from the Skele
variable is assigned an initial value when it is declared.

6. Explain the operation of the procedure PlaceShip.

7. The skeleton program utilises the variable Board.

   (a) Describe the data structure held by Board.

   (b) How is the data stored and used in this structure?

8. State the name of an identifier for:

   (a) A subroutine that contains a nested loop

   (b) A user-defined data type

   (c) A variable that stores text

   (d) A constant

   (e) A library function with exactly one parameter that returns an integer v

9. Look at the procedure PrintBoard.

   (a) What lines of code print the column headings?

   ....

   ....

   ....

   (b) What is the advantage of this procedure over 'hard-coding'?

   ....

   ....

   ....

10. This question is in relation to the routines PlaceRandomShips and LoadGa

    These routines both use a local variable called Row. What are local variable

    to these routines what is an advantage of utilising local variables?

    ....

    ....

    ....

    ....

11. The procedure PrintBoard utilises a For loop, whereas the Main procedure u

    What is the difference between a For loop and a Do Until loop?

    ....

    ....

    ....

    ....

    ....

12. SetUpShips is a procedure, whereas GetMainMenuChoice is a function.

    Describe the difference between a procedure and a function.

    ....

    ....

    ....

13. What is the purpose of the following line?

```
Using FileReader As StreamReader = New StreamReader(Filename)
```

14. What is the purpose of these lines?

```
Line = FileReader.ReadLine()
For Column = 0 To 9
    Board(Row, Column) = Line(Column)
Next
```

15. The LoadGame procedure uses the file Training.txt by default.

    (a) What would happen to the program if Training.txt did not exist?

    (b) Describe how we would change the program to solve this.

Programming Theory Questio

These questions refer to the Preliminary Material and require you to loa
but do not require any additional programming.

1. State the name of an identifier for:

   (a) An array or list variable

   (b) A subroutine that has five parameters

   (c) A variable that is used to store a whole number

   (d) A subroutine that returns one or more values

   (e) A variable that stores a Boolean value

2. Look at the function ValidateBoatPosition.

   What is the purpose of the variable Orientation?

3 What data is stored for each ship?

4. Look at the procedure PlayGame.

   What is the purpose of the Do Until loop?

5. Give an example of a declaration and assignment statement from the Skele variable is assigned an initial value when it is declared.

6. Explain the operation of the procedure PlaceShip.

7. The skeleton program utilises the variable Board.

   (a) Describe the data structure held by Board.

   (b) How is the data stored and used in this structure?

8. State the name of an identifier for:

   (a) A subroutine that contains a nested loop

   (b) A user-defined data type

   (c) A variable that stores text

   (d) A constant

   (e) A library function with exactly one parameter that returns an integer v

9. Look at the procedure PrintBoard.

   (a) What lines of code print the column headings?

   (b) What is the advantage of this method over 'hard-coding'?

10. This question is in relation to the routines PlaceRandomShips and LoadGa

These routines both use a local variable called Row. What are local variables to these routines what is an advantage of utilising local variables?

11. The procedure PrintBoard utilises a For loop, whereas the Main procedure u

What is the difference between a For loop and a Do Until loop?

12. SetUpShips is a procedure, whereas GetMainMenuChoice is a function.

Describe the difference between a procedure and a function.

13. What is the purpose of the following line?

```
Using FileReader As StreamReader = New StreamReader(Filename)
```

14. What is the purpose of these lines?

```
Line = FileReader.ReadLine()
For Column = 0 To 9
    Board(Row, Column) = Line(Column)
Next
```

15. The LoadGame procedure uses the file Training.txt by default.

(a) What would happen to the program if Training.txt did not exist?

(b) Describe how we would change the program to solve this.

Programming Exercises

The following require you to open the skeleton program and make modifications. Th
and illustrate how you should prepare your answer

Question 1

This question refers to GetRowColumn.

It is currently possible to fire at coordinates that are off the board, crashing the g
that this is not possible. If a square off the board is targeted, the message: 'Sorry
Please select again.' should be displayed and the user prompted to re-enter.

Evidence you need to provide

- Your amended SOURCE CODE PROGRAM for GetRowColumn
- SCREEN CAPTURE(S) of testing a shot at column 14 row -8

Question 2

This question refers to PlayGame.

It is currently possible to fire at every square in order until you find every ship. A
only has 20 torpedoes. The number of torpedoes should decrease by 1 after eve
screen. When the number of torpedoes reaches 0, the message 'GAME OVER! Yo
displayed and the game should end.

Evidence you need to provide

- Your amended SOURCE CODE PROGRAM for PlayGame.
- SCREEN CAPTURE(S) of testing showing the number of torpedoes going d
message

Question 3

This question refers to DisplayMenu and Main.

Alter the menu so that option 3 is also displayed between options 2 and 9.
The menu should display '3. Load saved game'.
If option 3 is selected, that program should display 'OPTION 3 EXECUTED'.

Evidence you need to provide

- Your amended SOURCE CODE PROGRAM for DisplayMenu
- SCREEN CAPTURE(S) of testing

Question 4

This question refers to Main.

Alter the procedure so that if the user enters 9 they are prompted with an 'Are yo
respond Y will the program quit.

Evidence you need to provide

- Your amended SOURCE CODE PROGRAM for Main
- SCREEN CAPTURE(S) of testing

Question 5

This question refers to Main.

Option 3 currently just displays a message. Amend it so that it prompts the user
this file and plays the game.

Evidence you need to provide

- Your amended SOURCE CODE PROGRAM for Main
- SCREEN CAPTURE(S) of testing using the filename 'Training.txt'

Question 6

Create a procedure called SaveGame. It should accept the board as a parameter
variable called filename.

It should then save the current state of the board to a text file named the value o
format as Training.txt.

Evidence you need to provide

- Your SOURCE CODE PROGRAM for SaveGame

Question 7

This question refers to PlayGame.

After a player has made a move they should be prompted: 'Do you want to save
If the player enters Y they should then be prompted for a filename and the gam
created in Question 6.

Evidence you need to provide

- Your amended SOURCE CODE PROGRAM for PlayGame
- SCREEN CAPTURE(S) of loading a game saved by the user

Question 8

This question refers to multiple sections of the skeleton code.

Create a menu option '4. Board Test'. It will set up a board and then display the generated board (revealing the location of the ships). After the board has been return to the main menu. A procedure called RealBoard (similar to PrintBoard) s board.

Evidence you need to provide

- Your amended sections of SOURCE CODE PROGRAM highlighting your ch
- SCREEN CAPTURE(S) of testing

Question 9

This question refers to multiple sections of the skeleton code.

A new ship has joined the fleet called a Frigate. It has a length of 3. Amend the placed in addition to the original ships when option 1 or 4 is selected. 'F' will re

Evidence you need to provide

- Your amended sections of the SOURCE CODE PROGRAM highlighting you
- SCREEN CAPTURE(S) using menu option 4 to show the Frigate

Question 10

This question refers to MakePlayerMove.

When a player misses, a radar scan of the adjacent cells should be performed. If section of ship, the message 'Enemy Near!' should be displayed. If not, the mess displayed. You should create a function called RadarScan that returns a Boolea enemy near).

Evidence you need to provide

- Your amended SOURCE CODE PROGRAM for MakePlayerMove
- Your new SOURCE CODE PROGRAM for RadarScan
- SCREEN CAPTURE(S) showing both types of radar scan message

Question 11

This question refers to PlayGame.

When a ship is hit its type must be displayed, e.g.:
Hit Aircraft Carrier at (8,6)

Evidence you need to provide

- Your amended sections of the SOURCE CODE PROGRAM highlighting you
- SCREEN CAPTURE(S) of a successful hit and the message

Question 12

This question refers to PlaceShip, ValidateBoatPosition and PlaceRandomShips.

Amend the program so that all ships can be placed diagonally down and to the
board or overlap with other ships, e.g.:

B			
	B		
		B	
			B

Evidence you need to provide

- Your amended sections of the SOURCE CODE PROGRAM highlighting you
- SCREEN CAPTURE(S) of a board generated by option 4 showing at least o

Question 13

This question refers to MakePlayerMove.

Amend the program so that if a ship is hit its size is reduced by 1.
A message will then display how many pieces of the ship are left to hit.

e.g.
Hit Battleship at (5,3)
There are 3 pieces of Battleship left

When the size reaches zero an additional message should say that the ship has

e.g.
Hit Battleship at ([illegible],6)
There are 0 pieces of Battleship left
YOU SANK THE BATTLESHIP

Evidence you need to provide

- Your amended sections of the SOURCE CODE PROGRAM highlighting you
- SCREEN CAPTURE(S) of a ship being sunk

Question 14

This question refers to multiple sections of the skeleton code.

A new menu option needs to be added: '5. Manually place ships'.

When selected the user will be prompted for the starting square and orientation program will then check whether this location is valid using ValidateBoatPositio selected, a message will confirm that the ship is placed and then place the ship u

e.g. Aircraft Carrier successfully placed at (1,3)

If ValidateBoatPosition returns false an error message will be displayed.
e.g. Invalid location. Please choose again.

After each ship has been placed, the Rea[illegible] procedure should display the po

When all ships are placed the game should begin.

Evidence you need to provide

- Your amended sections of the SOURCE CODE PROGRAM highlighting you
- SCREEN CAPTURE(S) showing the board before and after the submarine is

Question 15

This question refers to multiple sections of the skeleton code.

Create a variable to store the current player's score. Everybody starts at 0. Add score is better.

Create a user-defined data structure (similar to ship) called score.
It should contain a name and a score in suitable data types.

An array/list of five scores will store the scores.

Create a procedure (similar to SetUpBoard and SetUpShips) called SetUpScores. with the following data. It should only do this once when the program is first ru

George	17
Paul	19
John	23
Ringo	25
Bryan	35

Create a menu option '6. Display high score table' that executes a suitable proc

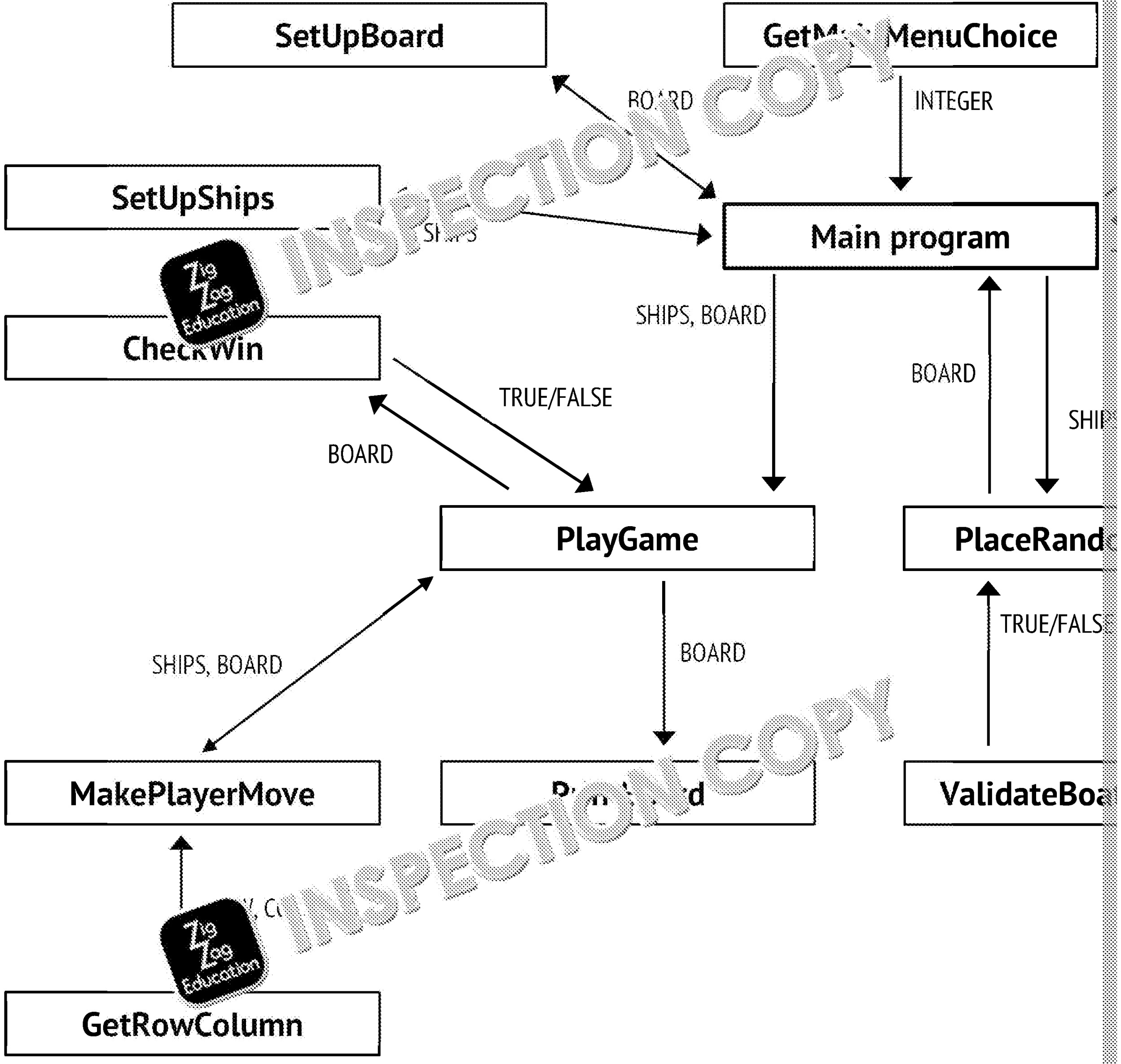
Create a procedure to bubble sort the high-score table called BubSortScores.

If a player scores less than somebody on the table (remember that a lower score on the table will be replaced with their name (you will need to prompt for this) a using BubSortScores.

Evidence you need to provide

- Your amended sections of the SOURCE CODE PROGRAM highlighting you
- SCREEN CAPTURE(S) showing the table being displayed before and after a

Structure Chart (Solution)



Programming Theory Questions (Answers)

Q	Marking Guidance
1a	Ships / Board
1b	ValidateBoatPosition
1c	Row / Column / HorV / MenuOption
1d	GetRowColumn / ValidateBoatPosition / CheckWin / GetMainMenuChoice
1e	Valid / GameWon
2	To store whether the boat should be vertically or horizontally positioned (1 ma board (1 mark)
3	Name (1 mark), Size (1 mark)
4	To ensure that the board is reprinted (1 mark) and the user input requested ag (1 mark) otherwise the game is not yet won (1 mark)
5	Dim GameWon As Boolean = False
6	To check whether the ship can be placed on the board (1 mark) by ensuring th edge of the board (1 mark) or run across another ship (1 mark). A value of true will only be returned if neither of these situations is the case (1
7a	Character array / char array / 2D array of characters
7b	Any three points (1 mark each): • Two-dimensional array • 10-by-10 array • One dimension for the column • One dimension for the row • A row,column / x,y value is used to refer to each element
8a	LoadGame / PlaceRandomShips
8b	TShip (reject Ships; this is an array)
8c	Line (reject TrainingGame; this is a constant)
8d	TrainingGame
8e	StreamReader

9a	1 mark for print line, 2 marks for For loop:
	For Column = 0 To 9
	        Console.Write(" " & Column & " ")
	    Next

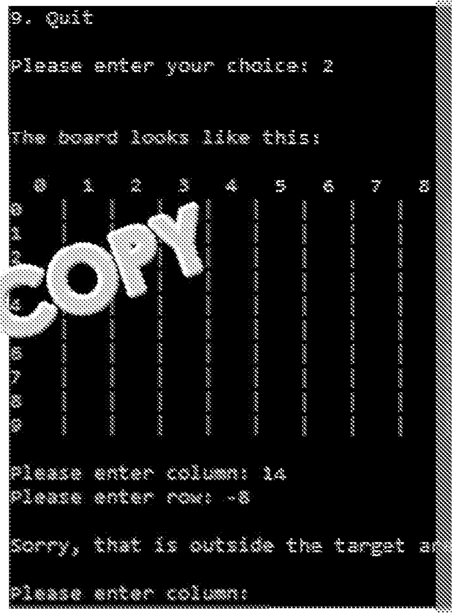
9b	It is easier to modify the code (1 mark), it allows many lines of code to be con (1 mark)
10	Local variable: stores a value for only that particular routine. The value is lost w (1 mark) Both routines can use the <u>same variable names</u> to traverse the array <u>without a</u> (2 marks for showing understanding of underlined words; 1 mark for partial un
11	A For loop repeats a set number of times (1 mark) and the number of times is k before the loop starts (1 mark). A Do Until loop repeats an unknown number of times (1 mark) while a certain c

Q	Marking Guidance
12	A procedure is a routine called by the program which performs a set of actions A function is a routine called within an expression which returns a result (1 ma
13	The data stored in the file is loaded up into an Object called FileReader.
14	Reads a line of the training game file (1 mark), then for each column (1 mark) s individual characters (1 mark) and assigns them to the correct position on the b
15a	It would crash
15b	A try catch (1 mark) should be used to catch the error (1 mark) and then display (1 mark).

Programming Exercises (Solutions)

Question	Answer
1	***GetRowColumn***

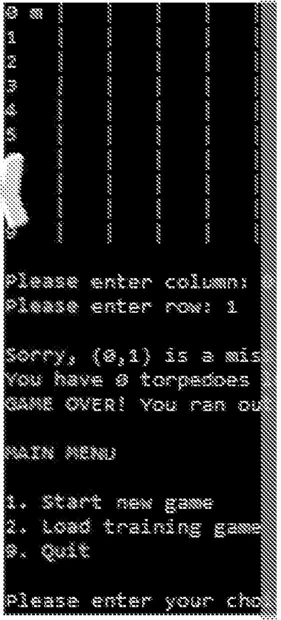
```
Sub GetRowColumn(ByRef Row As Integer, ByRef Column As Integer)
    Do
      Console.WriteLine()
      Console.Write("Please enter column: ")
      Column = Console.ReadLine()
      Console.Write("Please enter row: ")
      Row = Console.ReadLine()
      Console.WriteLine()
      If ((Row < 0) Or (Row > 9) Or (Column < 0) Or (Column > 9)) Then
        Console.WriteLine("Sorry, that is outside the target area.  Please select again.")
      End If
    Loop Until ((Row >= 0) And (Row <= 9) And (Column >= 0) And (Column <= 9))
  End Sub
```



```
9. Quit

Please enter your choice: 2


The board looks like this:

  0   1   2   3   4   5   6   7   8
0   |   |   |   |   |   |   |   |
1   |   |   |   |   |   |   |   |
2   |   |   |   |   |   |   |   |
3   |   |   |   |   |   |   |   |
4   |   |   |   |   |   |   |   |
5   |   |   |   |   |   |   |   |
6   |   |   |   |   |   |   |   |
7   |   |   |   |   |   |   |   |
8   |   |   |   |   |   |   |   |
9   |   |   |   |   |   |   |   |

Please enter column: 14
Please enter row: -8

Sorry, that is outside the target ar

Please enter column:
```

Question	Answer
2	***PlayGame***

```
Sub PlayGame(ByVal Board(,) As Char, ByVal Ships() As TShip)
    Dim GameWon As Boolean = False
    Dim Torpedoes As Integer = 2
    Do
      PrintBoard(Board)
      MakePlayerMove(Board, Ships)
      Torpedoes = Torpedoes - 1
      Console.WriteLine("You have " & Torpedoes & " torpedoes left")
      GameWon = CheckWin(Board)
      If GameWon Then
        Console.WriteLine("All ships sunk!")
        Console.WriteLine()
      End If
      If Torpedoes = 0 Then
        Console.WriteLine("GAME OVER! You ran out of ammo")
        Console.WriteLine()
      End If
    Loop Until GameWon Or Torpedoes = 0
  End Sub
```



```
0 m |   |   |   |   |
1   |   |   |   |   |
2   |   |   |   |   |
3   |   |   |   |   |
4   |   |   |   |   |
5   |   |   |   |   |
    |   |   |   |   |
    |   |   |   |   |
    |   |   |   |   |
9   |   |   |   |   |

Please enter column:
Please enter row: 1

Sorry, (0,1) is a mis
You have 0 torpedoes
GAME OVER! You ran ou

MAIN MENU

1. Start new game
2. Load training game
9. Quit

Please enter your cho
```

Question	Answer
3	see below

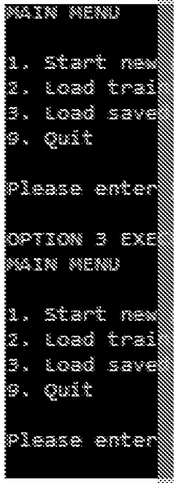
3

DisplayMenu

```
    ...
    Console.WriteLine("1. Start new game")
    Console.WriteLine("2. Load training game")
    Console.WriteLine("3. Load saved game")
    Console.WriteLine("9. Quit")
    ...
```

Main

```
      ...
      If MenuOption = 1 Then
        PlaceRandomShips(Board, Ships)
        PlayGame(Board, Ships)
      ElseIf MenuOption = 2 Then
        LoadGame(TrainingGame, Board)
        PlayGame(Board, Ships)
      ElseIf MenuOption = 3 Then
        Console.WriteLine("OPTION 3 EXECUTED")
      End If
      ...
```

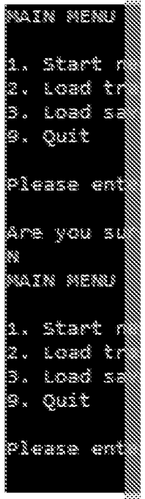


```
MAIN MENU

1. Start new
2. Load trai
3. Load save
9. Quit

Please enter

OPTION 3 EXE
MAIN MENU

1. Start new
2. Load trai
3. Load save
9. Quit

Please enter
```

4

Main

```
      ...
      If MenuOption = 1 Then
        PlaceRandomShips(Board, Ships)
        PlayGame(Board, Ships)
      ElseIf MenuOption = 2 Then
        LoadGame(TrainingGame, Board)
        PlayGame(Board, Ships)
      ElseIf MenuOption = 3 Then
        Console.WriteLine("OPTION 3 EXECUTED")
      ElseIf MenuOption = 9 Then
        Console.WriteLine("Are you sure (Y/N) ?")
        If Console.ReadLine <> "Y" Then
          MenuOption = 0
        End If
      End If
      ...
```



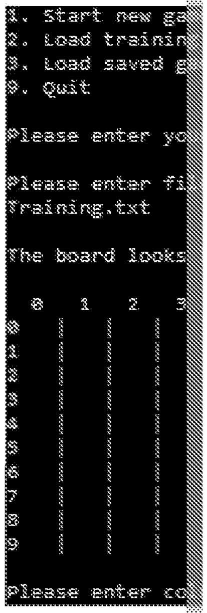
```
MAIN MENU

1. Start n
2. Load tr
3. Load sa
9. Quit

Please ent

Are you su
N
MAIN MENU

1. Start n
2. Load tr
3. Load sa
9. Quit

Please ent
```

Question | **Answer**

5

Main

```
    ...
    If MenuOption = 1 Then
      PlaceRandomShips(Board, Ships)
      PlayGame(Board, Ships)
    ElseIf MenuOption = 2 Then
      LoadGame(TrainingGame, Board)
      PlayGame(Board, Ships)
    ElseIf MenuOption = 3 Then
      Dim FileName As String
      Console.WriteLine("Please enter file name:")
      FileName = Console.ReadLine()
      LoadGame(FileName, Board)
      PlayGame(Board, Ships)
    ...
```

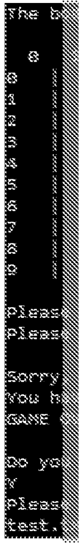
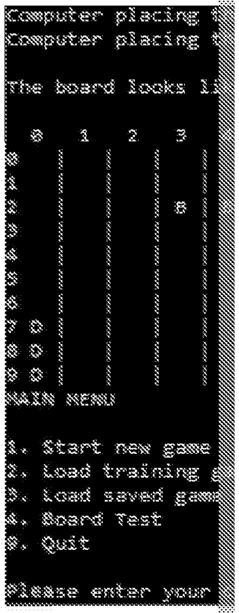


```
1. Start new ga
2. Load trainin
3. Load saved g
9. Quit

Please enter yo

Please enter fi
Training.txt

The board looks

  0   1   2   3
0    |   |   |
1    |   |   |
2    |   |   |
3    |   |   |
4    |   |   |
5    |   |   |
6    |   |   |
7    |   |   |
8    |   |   |
9    |   |   |

Please enter co
```

6

SaveGame

```
Sub SaveGame(ByVal Filename As String, ByVal Board(,) As Char)
  Dim Row As Integer
  Dim Column As Integer
  Dim Line As String = ""
  Using FileWriter As StreamWriter = New StreamWriter(Filename)
    For Row = 0 To 9
      For Column = 0 To 9
        Line = Line & Board(Row, Column)
      Next
      FileWriter.WriteLine(Line)
      Line = ""
    Next
  End Using
End Sub
```

Question	Answer
7	***PlayGame*** ... If Torpedoes = 0 Then Console.WriteLine("GAME OVER! You ran out of torpedoes") Console.WriteLine() End If **Console.WriteLine("Do you want to save the game (Y,N)?")** **If Console.ReadLine() = "Y" Then** **Console.WriteLine("Please enter file name:")** **Dim FName As String** **FName = Console.ReadLine()** **SaveGame(FName, Board)** **End If** Loop Until GameWon Or Torpedoes = 0 End Sub  The b 0 0 1 2 3 4 5 6 7 8 9 Pleas Pleas Sorry You h GAME (Do yo Y Pleas test.
8	***DisplayMenu*** ... Console.WriteLine("1. Start new game") Console.WriteLine("2. Load training game") Console.WriteLine("3. Load saved game") **Console.WriteLine("4. Board Test")** Console.WriteLine("9. Quit") ... ***Main*** ... ElseIf MenuOption = 3 Then Dim FileName As String Console.WriteLine("Please enter file name:") FileName = Console.ReadLine() LoadGame(FileName, Board) PlayGame(Board, Ships) **ElseIf MenuOption = 4 Then** **PlaceRandomShips(Board, Ships)** **RealBoard(Board)** ...  Computer placing t Computer placing t The board looks li 0 1 2 3 0 | | | | 1 | | | | 2 | | | B | 3 | | | | 4 | | | | 5 | | | | 6 | | | | 7 D | | | | 8 D | | | | 9 D | | | | MAIN MENU 1. Start new game 2. Load training g 3. Load saved game 4. Board Test 9. Quit Please enter your

RealBoard

```
Sub RealBoard(ByVal Board(,) As Char)
   Dim Row As Integer
   Dim Column As Integer
   Console.WriteLine()
   Console.WriteLine("The board looks like this: ")
   Console.WriteLine()
   Console.Write(" ")
   For Column = 0 To 9
      Console.Write(" " & Column & "  ")
   Next
   Console.WriteLine()
   For Row = 0 To 9
      Console.Write(Row & " ")
      For Column = 0 To 9
         If Board(Row, Column) = "-" Then
            Console.Write(" ")
         'ElseIf Board(Row, Column) = "A" Or Board(Row, Column) = "B" Or Board(Row, Colu
            Column) = "D" Or Board(Row, Column) = "P" Then
            'Console.Write(" ")
         Else
            Console.Write(Board(Row, Column))
         End If
         If Column <> 9 Then
            Console.Write(" | ")
         End If
      Next
      Console.WriteLine()
   Next
End Sub
```

Question	Answer
9	

Main

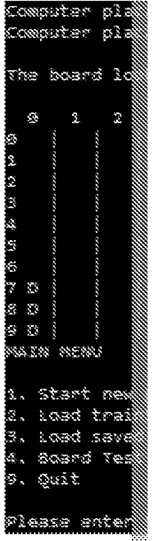
```
Sub Main()
   Dim Board(9, 9) As Char
   Dim Ships(5) As TShip
   ...
```

SetUpShips

```
 ...
   Ships(4).Name = "Patrol Boat"
   Ships(4).Size = 2
   Ships(5).Name = "Frigate"
   Ships(5).Size = 3
End Sub
```

CheckWin

```
   ...
   For Row = 0 To 9
      For Column = 0 To 9
         If Board(Row, Column) = "A" Or Board(Row, Column) = "B" Or Board(Row, Column
         "D" Or Board(Row, Column) = "P" Or Board(Row, Column) = "F" Then
            Return False
         End If
      ...
```



```
Computer pla
Computer pla

The board lo

  0   1   2
0    |   |
1    |   |
2    |   |
3    |   |
4    |   |
5    |   |
6    |   |
7 D  |   |
8 D  |   |
9 D  |   |
MAIN MENU

1. Start new
2. Load trai
3. Load save
4. Board Tes
9. Quit

Please enter
```

PrintBoard

```
   ...
   For Row = 0 To 9
      Console.Write(Row & " ")
      For Column = 0 To 9
         If Board(Row, Column) = "-" Then
            Console.Write(" ")
         ElseIf Board(Row, Column) = "A" Or Board(Row, Column) = "B" Or Board(Row, Col
         Column) = "D" Or Board(Row, Column) = "P" Or Board(Row, Column) = "F" Then
            Console.Write(" ")
         Else
            Console.Write(Board(Row, Column))
         End If
         ...
```

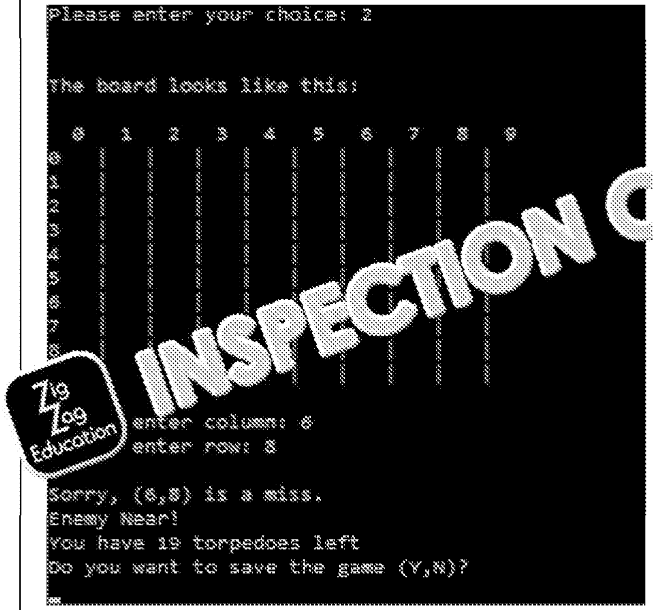
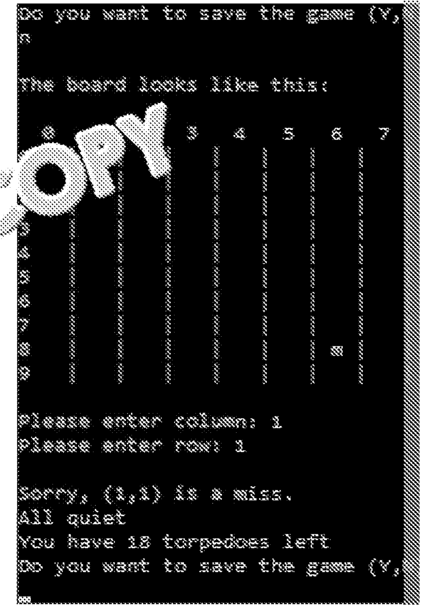
Question	Answer
10	

MakePlayerMove

```
    ...
    If Board(Row, Column) = "m" Or Board(Row, Column) = "h" Then
        Console.WriteLine("Sorry, you have already fired at that square (" & Column & "," & 
    ElseIf Board(Row, Column) = "-" Then
        Console.WriteLine("Sorry, (" & Column & "," & Row & ") is a miss.")
        If (RadarScan(Board, Row, Column) = True) Then
            Console.WriteLine("Enemy Near!")
        Else
            Console.WriteLine("All quiet")
        End If
        Board(Row, Column) = "m"
    Else
    ...
```

RadarScan

```
  Function RadarScan(ByVal Board(,) As Char, ByVal Row As Integer, ByVal Column As Int
    For ColumnScan = Column - 1 To Column + 1
        For RowScan = Row - 1 To Row + 1
            If (ColumnScan > 9) Or (ColumnScan < 0) Or (RowScan > 9) Or (RowScan < 0) The
                'do nothing (outside of board)
            Else
                If ((Board(RowScan, ColumnScan) <> "-") And (Board(RowScan, ColumnScan) <
                <> "h")) Then
                    Return True
                End If
            End If
        Next
    Next
    Return False
  End Function
```

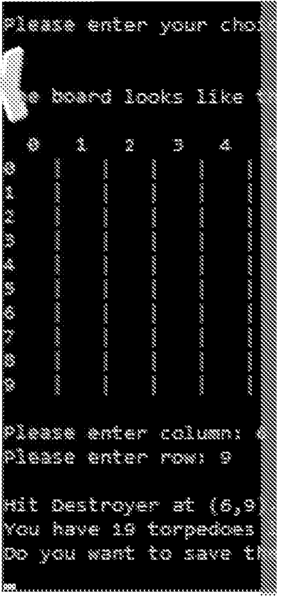
	 

```
Please enter your choice: 2


The board looks like this:

  0   1   2   3   4   5   6   7   8   9
0   |   |   |   |   |   |   |   |   |
1   |   |   |   |   |   |   |   |   |
2   |   |   |   |   |   |   |   |   |
3   |   |   |   |   |   |   |   |   |
4   |   |   |   |   |   |   |   |   |
5   |   |   |   |   |   |   |   |   |
6   |   |   |   |   |   |   |   |   |
7   |   |   |   |   |   |   |   |   |
8   |   |   |   |   |   |   |   |   |
9   |   |   |   |   |   |   |   |   |

Please enter column: 6
Please enter row: 8

Sorry, (6,8) is a miss.
Enemy Near!
You have 19 torpedoes left
Do you want to save the game (Y,N)?
```

```
Do you want to save the game (Y,
n

The board looks like this:

  0   1   2   3   4   5   6   7
0   |   |   |   |   |   |   |
1   |   |   |   |   |   |   |
2   |   |   |   |   |   |   |
3   |   |   |   |   |   |   |
4   |   |   |   |   |   |   |
5   |   |   |   |   |   |   |
6   |   |   |   |   |   |   |
7   |   |   |   |   |   |   |
8   |   |   |   |   |   | m |
9   |   |   |   |   |   |   |

Please enter column: 1
Please enter row: 1

Sorry, (1,1) is a miss.
All quiet
You have 18 torpedoes left
Do you want to save the game (Y,
```

Question	Answer
11	***MakePlayerMove***

```
    ...
    If (RadarScan(Board, Row, Column) = True) Then
      Console.WriteLine("Enemy Near!")
    Else
      Console.WriteLine("Al[illegible]")
    End If
    Board(Row, Column) = "m"
  Else
    Dim ShipName As String = ""
    Select (Board(Row, Column))
      Case "A"
        ShipName = "Aircraft Carrier"
      Case "B"
        ShipName = "Battleship"
      Case "S"
        ShipName = "Submarine"
      Case "D"
        ShipName = "Destroyer"
      Case "P"
        ShipName = "Patrol Boat"
      Case Else
    End Select
    Console.WriteLine("Hit " & ShipName & " at (" & Column & "," & Row & ").")
    Board(Row, Column) = "h"
  End If
End Sub
```

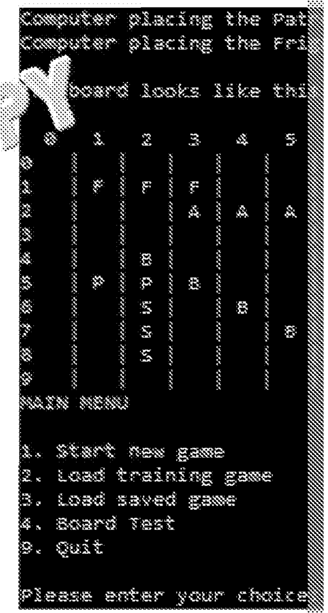


```
Please enter your cho

The board looks like t

  0   1   2   3   4
0   |   |   |   |   |
1   |   |   |   |   |
2   |   |   |   |   |
3   |   |   |   |   |
4   |   |   |   |   |
5   |   |   |   |   |
6   |   |   |   |   |
7   |   |   |   |   |
8   |   |   |   |   |
9   |   |   |   |   |

Please enter column: 6
Please enter row: 9

Hit Destroyer at (6,9
You have 19 torpedoes
Do you want to save th
```

Question	Answer
12	see below

PlaceRandomShips

```
...
For Each Ship In Ships
   Valid = False
   While Not Valid
      Row = Int(Rnd() * 10)
      Column = Int(Rnd() * 10)
      HorV = Int(Rnd() * 3)
      If HorV = 0 Then
         Orientation = "v"
      ElseIf (HorV = 1) Then
         Orientation = "d"
      Else
         Orientation = "h"
      End If
      ...
```



```
Computer placing the Pat
Computer placing the Fri

board looks like thi

   0    1    2    3    4    5
0     |    |    |    |    |
1     | F  | F  | F  |    |
2     |    |    | A  | A  | A
3     |    |    |    |    |
4     |    | B  |    |    |
5     | P  | P  | B  |    |
6     |    | S  |    | B  |
7     |    | S  |    |    | B
8     |    | S  |    |    |
9     |    |    |    |    |
MAIN MENU

1. Start new game
2. Load training game
3. Load saved game
4. Board Test
9. Quit

Please enter your choice
```

PlaceShip

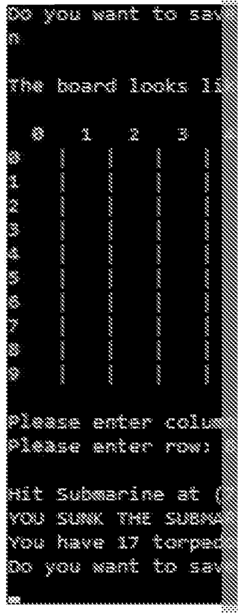
```
...
If Orientation = "v" Then
   For Scan = 0 To Ship.Size - 1
      Board(Row + Scan, Column) = Ship.Name(0)
   Next
ElseIf Orientation = "h" Then
   For Scan = 0 To Ship.Size - 1
      Board(Row, Column + Scan) = Ship.Name(0)
   Next
Else
   For Scan = 0 To Ship.Size - 1
      Board(Row + Scan, Column + Scan) = Ship.Name(0)
   Next
End If
End Sub
```

ValidateBoatPosition

```
Function ValidateBoatPosition(ByVal Board(,) As Char, ByVal Ship As TShip, ByVal Row As I
Integer, ByVal Orientation As Char)
    Dim Scan As Integer
    If (Orientation = "v" Or Orientation = "d") And Row + Ship.Size > 10 Then
      Return False
    ElseIf (Orientation = "h" Or Orientation = "l") And Column + Ship.Size > 10 Then
      Return False
    Else
      If Orientation = "v" Then
        For Scan = 0 To Ship.Size - 1
          If Board(Row + Scan, Column) <> "-" Then
            Return False
          End If
        Next
      ElseIf (Orientation = "h") Then
        For Scan = 0 To Ship.Size - 1
          If Board(Row, Column + Scan) <> "-" Then
            Return False
          End If
        Next
      Else
        For Scan = 0 To Ship.Size - 1
          If Board(Row + Scan, Column + Scan) <> "-" Then
            Return False
          End If
        Next
      End If
    End If
    Return True
  End Function
     for (int scan = 0; scan < ship.size; scan++){
     if(Board[row + scan, column + scan] != '-'){
       return false;

  }
  return true;
 }
```

Question	Answer
13	***MakePlayerMove***

```
      ...
      If (RadarScan(Board, Row, Column) = True) Then
         Console.WriteLine("Enemy Near!")
      Else
         Console.WriteLine("[illegible]")
      End If
      Board(Row, Column) = "m"
   Else
      Dim ShipNum As Integer
      For i = 0 To Ships.Length - 1
         If Ships(i).Name(0) = Board(Row, Column) Then
            ShipNum = i
         End If
      Next
      Console.WriteLine("Hit " & Ships(ShipNum).Name & "
      at (" & Column & "," & Row & ").")
      Board(Row, Column) = "h"
      Ships(ShipNum).Size = Ships(ShipNum).Size - 1
      If Ships(ShipNum).Size = 0 Then
         Console.WriteLine("YOU SUNK THE " &
          Ships(ShipNum).Name.ToUpp[illegible]
      End If
   End If
End Sub
```



```
Do you want to sav
n

The board looks li

  0   1   2   3
0    |   |   |   |
1    |   |   |   |
2    |   |   |   |
3    |   |   |   |
4    |   |   |   |
5    |   |   |   |
6    |   |   |   |
7    |   |   |   |
8    |   |   |   |
9    |   |   |   |

Please enter colum
Please enter row:

Hit Submarine at (
YOU SUNK THE SUBMA
You have 17 torpec
Do you want to sav
```

Question	Answer
14	

DisplayMenu

```
Console.WriteLine("5. Manually place ships")
```

Main

```
    ElseIf MenuOption = 5 Then
        PlaceManualShips(Board, Ships)
        PlayGame(Board, Ships)
```

PlaceManualShips

```
    Sub PlaceManualShips(ByRef Board(,) As Char, ByVal Ships() As TShip)
        Dim Valid As Boolean
        Dim Row As Integer
        Dim Column As Integer
        Dim Orientation As Char
        For Each Ship In Ships
            Valid = False
            While Not Valid
                Console.WriteLine("Please enter row")
                Row = Console.ReadLine()
                Console.WriteLine("Please enter column")
                Column = Console.ReadLine()
                Console.WriteLine("Please enter orientation")
                Orientation = Console.ReadLine()
                Valid = ValidateBoatPosition(Board, Ship, Row, Column, Orientation)
                If Not Valid Then
                    Console.WriteLine("Invalid location. Please choose again")
                End If
            End While
            Console.WriteLine("Computer placing the " & Ship.Name)
            PlaceShip(Board, Ship, Row, Column, Orientation)
            [illegible]RealBoard(Board)
        Next
    End Sub
```

Question	Answer
15	

Score data structure

```
Structure TScore
    Dim Name As String
    Dim Score As Integer
End Structure
```

SetupScores

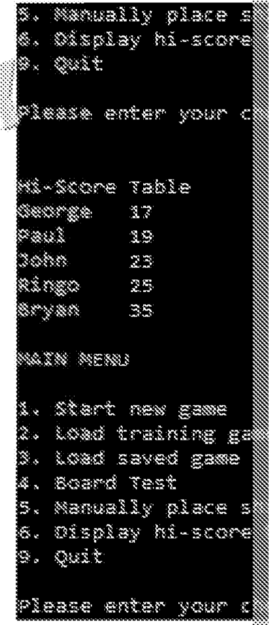
```
Sub SetUpScores(ByRef Scores() As TScore)
    Scores(0).Name = "George"
    Scores(0).Score = 17
    Scores(1).Name = "Paul"
    Scores(1).Score = 19
    Scores(2).Name = "John"
    Scores(2).Score = 23
    Scores(3).Name = "Ringo"
    Scores(3).Score = 25
    Scores(4).Name = "Bryan"
    Scores(4).Score = 35
End Sub
```

DisplayMenu

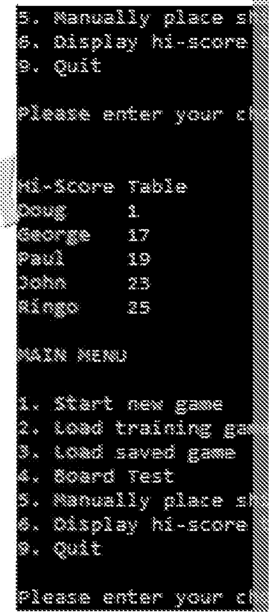
```
    Console.WriteLine("6. Display hi-score table")
```

BubSortScores

```
Sub BubSortScores(ByRef Scores() As TScore)
    Dim Changed As Boolean = True
    While (Changed)
        Changed = False
        For i = 0 To 3
            If (Scores(i).Score > Scores(i + 1).Score) Then
                Changed = True
                Dim TempScore As TScore = Scores(i + 1)
                Scores(i + 1) = Scores(i)
                Scores(i) = TempScore
            End If
        Next
    End While
End Sub
```



```
5. Manually place s
6. Display hi-score
9. Quit

Please enter your c


Hi-Score Table
George    17
Paul      19
John      23
Ringo     25
Bryan     35

MAIN MENU

1. Start new game
2. Load training ga
3. Load saved game
4. Board Test
5. Manually place s
6. Display hi-score
9. Quit

Please enter your c
```



```
5. Manually place sh
6. Display hi-score
9. Quit

Please enter your ch


Hi-Score Table
Doug      1
George    17
Paul      19
John      23
Ringo     25

MAIN MENU

1. Start new game
2. Load training ga
3. Load saved game
4. Board Test
5. Manually place sh
6. Display hi-score
9. Quit

Please enter your ch
```

DisplayHiScores

```
Sub DisplayHiScores(ByVal Scores() As TScore)
   Console.WriteLine()
   Console.WriteLine("Hi-Score Table")
   For Each Score In Scores
      Console.WriteLine(Score.Name + vbTab + Str(Score.Score))
   Next
   Console.WriteLine()
End Sub
```

Main

```
Sub Main()
   Dim Board(9, 9) As Char
   Dim Ships(5) As TShip
   Dim MenuOption As Integer
   Dim Scores(4) As TScore

   Do
      SetUpBoard(Board)
      SetUpShips(Ships)
      DisplayMenu()
      MenuOption = GetMainMenuChoice()
      If MenuOption = 1 Then
         PlaceRandomShips(Board, Ships)
         PlayGame(Board, Ships, Scores)
      ElseIf MenuOption = 2 Then
         LoadGame(TrainingGame, Board)
         PlayGame(Board, Ships, Scores)
      ElseIf MenuOption = 6 Then
        DisplayHiScores(Scores)
```

PlayGame

```
Sub PlayGame(ByRef Board(,) As Char, ByVal Ships() As TShip, ByRef Scores() As TScore)
   Dim GameWon As Boolean = False
   Dim Torpedoes As Integer = 20
   Dim PlayerScore = 0
   Do
      PrintBoard(Board)
      MakePlayerMove(Board, Ships)
```

```
        Torpedoes = Torpedoes - 1
        PlayerScore = PlayerScore + 1
        Console.WriteLine("You have " & Torpedoes & " torpedoes left")
        GameWon = CheckWin(Board)
        If GameWon Then
          Console.WriteLine("All ships sunk!")
          Console.WriteLine()
          If PlayerScore < Scores(4).Score Then
            Console.WriteLine("[illegible] hi-score")
            Scores(4).Score = PlayerScore
            Console.WriteLine("Please enter your name: ")
            Scores(4).Name = Console.ReadLine
            BubSortScores(Scores)
          End If
        End If
      {
        scores[4].score = score;
        console.println("Well done, you got a hi score");
        scores[4].name = console.readLine("Enter your name: ");
        BubSortScores(scores);
      }
    }
```

Further Modifications

Ideas for modifications	How to i

Name	

ZigZag Education supporting

AS AQA Computer Science Paper 1

Summer 2016: AQA WARSHIPS

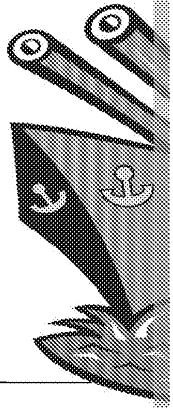
Electronic Answer Document (EAD)

Instructions

- Enter your name in the box at the top of this page
- Answer **all** questions by entering your answers into this document
- Remember to **save** this document regularly
- Save and print this document and any additional pages

- Answer **all** questions
- The marks available for each question are shown in brackets

- You will need:
  - □ access to a computer
  - □ access to a printer
  - □ access to appropriate software
  - □ electronic copies of the required skeleton code
  - □ EAD (Electronic Answer Document)



Total marks:

Programming Theory Questior

Answer all questions.
Remember to save this document regularly.

Q	Answer	
1	(a)	
	(b)	
	(c)	
	(d)	
	(e)	
2		
3		
4		
5		
6		
7	(a)	
	(b)	
8	(a)	
	(b)	
	(c)	
	(d)	
	(e)	
9	(a)	
	(b)	
10		
11		
12		
13		
14		
15	(a)	
	(b)	

Programming Exercises

Answer all questions.

Remember to save this document regularly.

Q	Answer
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	