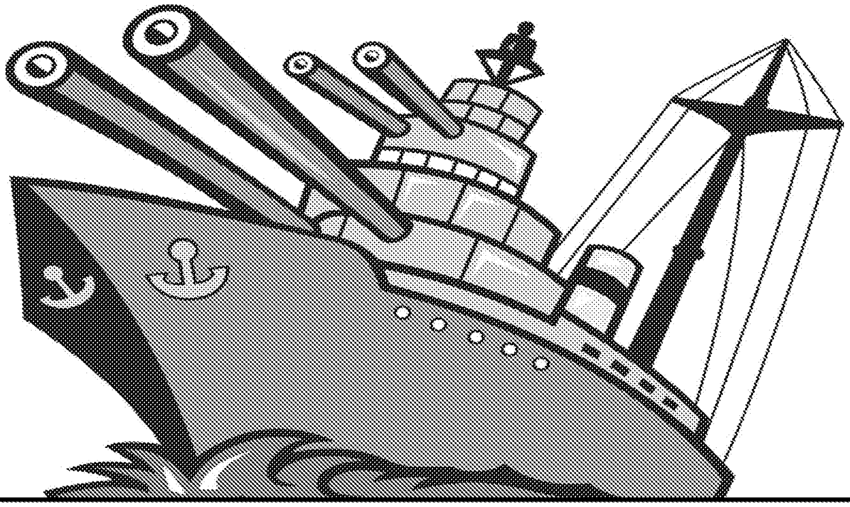
Zig Zag Education

Computer Science

AS | AQA | 7516

2015 specification
for the 2016 AS exam

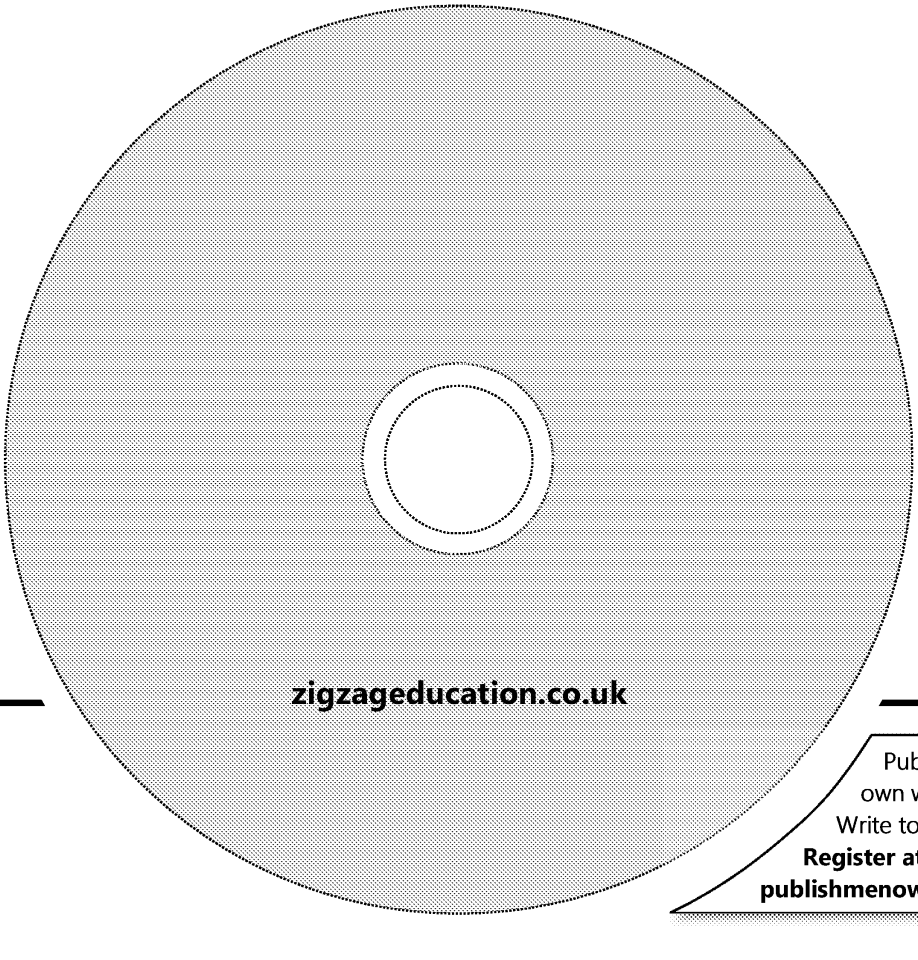


PAPER 1 EXAM RESOURCE PACK 2016

AQA WARSHIPS

for AS AQA Computer Science

PYTHON2



AS2/
6500

POD
6502

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 (PYTHON2) – for examination June 2016.

① **Pre-release Commentary** (for teachers)
A detailed overview of the skeleton program, describing all PYTHON2 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 PYTHON2 code are provided on the CD.

Answers and solutions for the structure chart activity, theory questions and programming exercises are provided from page 20 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 PY2 folder):

- **MODIFIED_PY2_CODE.txt** – text file containing the additional and/or modified program code as shown in the mark scheme for section ④ (from page 22).
- **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 b
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 grid. 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 filenam
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 (a
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 progr
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
Ships	An array/list	Stores the name and size of all the shi
Board	A two-dimensional array/list of one character strings	Stores the current state of the board
TRAININGGA	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 string 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

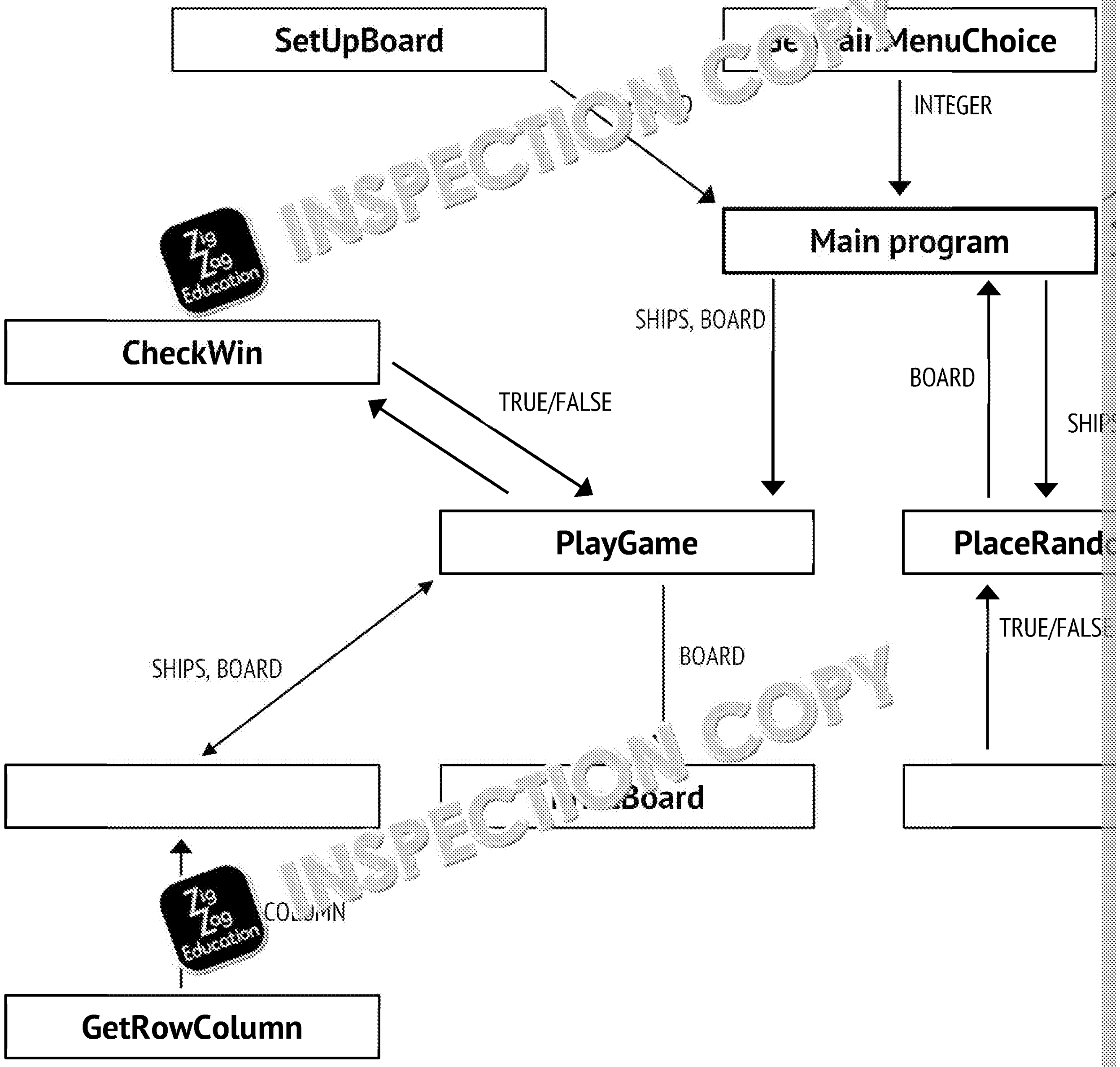
Routine	Description	
CheckWin Ⓕ	Receives: Board Returns: Boolean Called from: PlayGame	Checks every position in Board to Returns false if it finds a piece Returns true if it checks every pos
DisplayMen	Receives: nothing Returns: nothing Called from: main program	A simple procedure that prints op

Routine	Description	
GetMainMenuChoice Ⓕ	Receives: nothing Returns: integer Called from: main program	Handles the user's menu choice: 1. Prompts the user to ente 2. Returns that number
GetRowColumn Ⓕ	Receives: nothing Returns: Row, Column Called from: MakePlayerMove	1. Prompts the user for a co 2. Prompts the user for a ro 3. Returns both Row and Co
LoadGame Ⓟ	Receives: Filename, Board Returns: nothing Called from: main program	1. Reads the data containe 2. Then chops Line into ind board 3. Repeats for all 10 rows 4. 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
PlaceRandomShips Ⓟ	Receives: Board, Ships Returns: nothing Called from: main program	This procedure is not used when t It generates a random row, colum horizontally or vertically. It then uses the function Validate through that position, and that al the position is suitable, the boat i entered. This continues until a
PlaceShip Ⓟ	Receives: Board, Ship, Row, Column, Orientation Returns: nothing Called from: PlaceRandomShips	Places the ships on the board. Uses For loop that counts up to th counter is called Scan. Scan is ad the same). Scan is added to colum same). The board is populated in occupie Ship[0][0]).

Routine	Description
PlayGame Ⓟ	Receives: Board, Ships Returns: nothing Called from: main program Starts a game and k 1. Sets the Boolea 2. Starts a conditi while it is false 2.1. Displays t 2.2. Gets the p 2.3. Checks to displayed
PrintBoard	Receives: Board Returns: nothing Called from: PlayGame Displays the board 1. Starts off by dis 2. Next, a For loo 3. Nested For loo bottom) 3.1. Prints the 3.2. Second Fo 3.2.1. An e 3.2.2. A squ ships 3.2.3. Anyt 3.2.4. A sep to dis
SetUpBoard Ⓕ	Receives: nothing Returns: Board Called from: main program 1. Cycles through 1.1. Assigns al Some of these dash

Routine	Description	
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 it will 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 re returned.
Main program Ⓟ	1. Sets up the board 2. Declares a variable to store what menu option has been selected an 9) 3. Starts a conditional loop that continues until the user selects option 3.1. Populates board with data by calling SetUpBoard (this would re 3.2. Displays the menu by calling DisplayMenu 3.3. Calls GetMainMenuChoice to get the user's choice and stores it 3.4. If the user picks option 1: 3.4.1. The board is populated by the ships in random locations 3.4.2. The game is started 3.5. If the user picks option 2: 3.5.1. The board is populated from the training text file 3.5.2. The game is started	

AQA WARSH


SetUpBoard
MenuChoice
INTEGER
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 load
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 While 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 procedure that is passed 2 parameters

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

....

....

....

....

11. The procedure PrintBoard utilises a For loop, whereas the main program ut
What is the difference between a For loop and a While loop?

....

....

....

....

....

12. PrintBoard 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?

```
BoardFile = open(Filename, "r")
```

14. What is the purpose of these lines?

```
for Row in range(10):
   Line = BoardFile.readline()
   for Column in range(10):
     Board[Row][Column] = Line[Column]
```

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 While 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 procedure that is passed 2 parameters

   (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 LoadG

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

11. The procedure PrintBoard utilises a For loop, whereas the main program ut

What is the difference between a For loop and a While loop?

12. PrintBoard 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?

```
BoardFile = open(Filename, "r")
```

14. What is the purpose of these lines?

```
for Row in range(10):
   Line = BoardFile.readline()
   for Column in range(10):
     Board[Row][Column] = Line[Column]
```

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 the main program.

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 the main program.

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

Evidence you need to provide

- Your amended SOURCE CODE PROGRAM for the main program
- SCREEN CAPTURE(S) of testing

Question 5

This question refers to the main program.

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 the main program
- 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
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 game
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 MakePlayerMove.

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 P[illegible], validateBoatPosition and PlaceRandomShips.

Amend the [illegible]m 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 or

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 [illegible] message should say that the ship has b

e.g.
Hit Battlesh[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) called SetUpScores. It should popul data. It should only do this once when the program is first run.

George	17
Paul	19
John	23
Ringo	25
Bryan	35

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

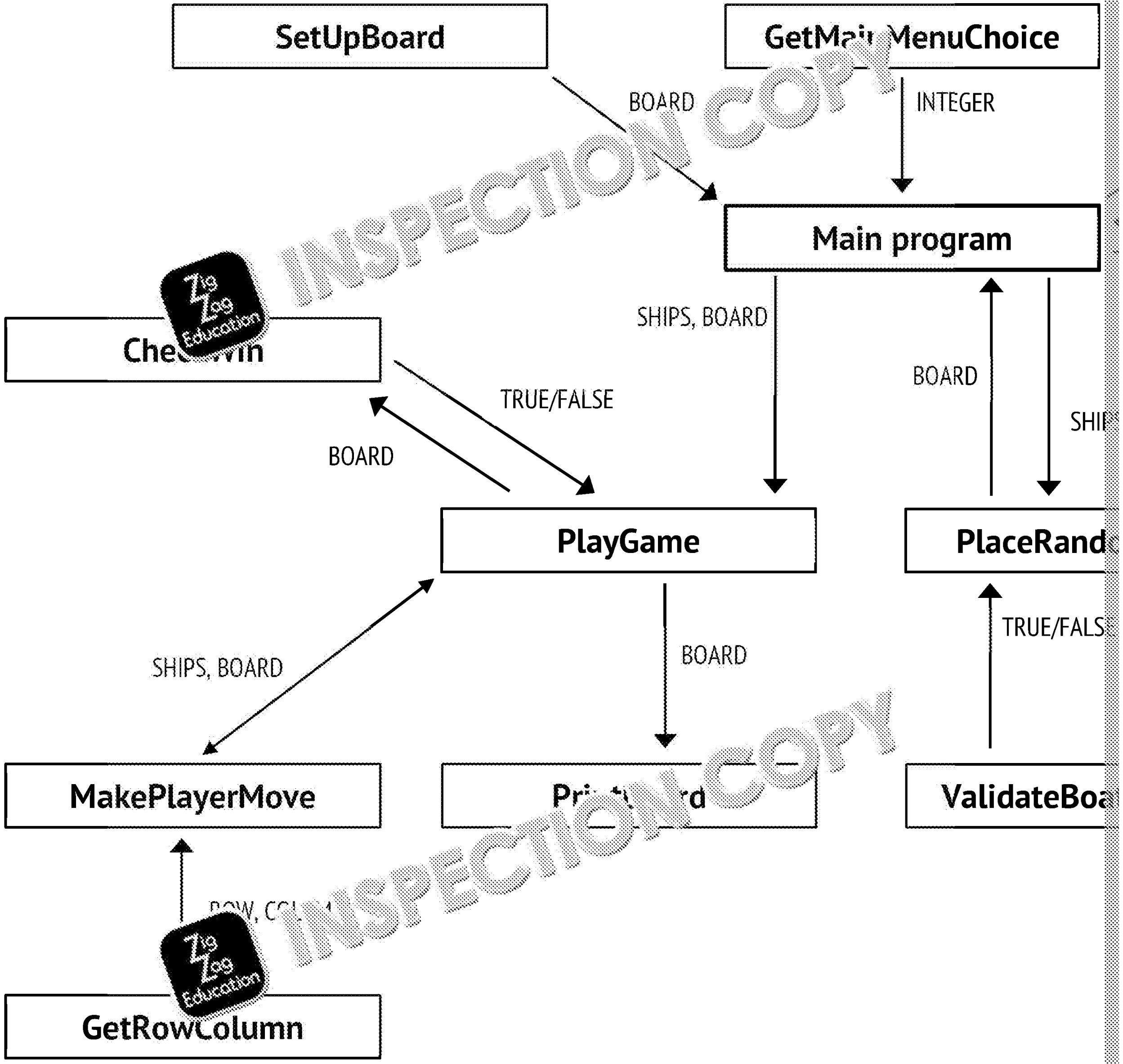
Create a procedure to bubble sort the high-score table called BubSortScores.

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 mar 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 aga (1 mark) while the game is not yet won (1 mark).
5	MenuOption = 0
6	To check whether the ship can be placed on the board (1 mark) by ensuring tha 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 3 from: Two-dimensional array (1 mark); 10-by-10 array (1 mark); One dimensi One dimension for the row (1 mark); A row,column / x,y value is used to refer to ea
8a	LoadGame / PlaceRandomShips
8b	PlayGame / LoadGame / MakePlayerMove
8c	Line (reject TRAININGGAME; this is a constant)
8d	TRAININGGAME
8e	Random

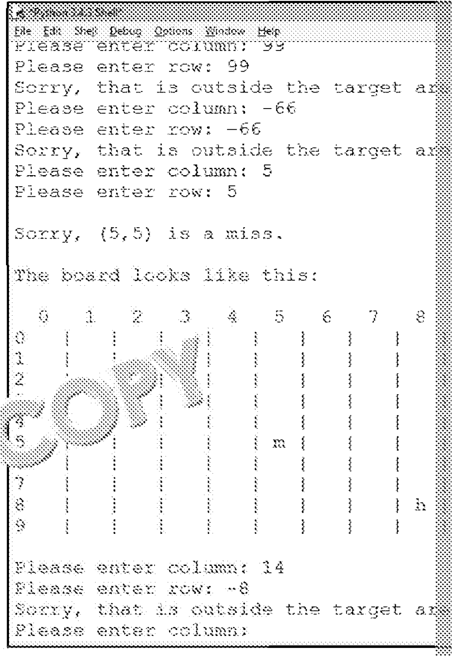
9a	1 mark for print line, 2 marks for For loop showing indent :
	
	for Column in range(10):
	        print " " + str(Column) + " ",

9b	It is easier to modify the game (1 mark), it allows many lines of code to be cond (1 mark).
10	Local variable: stores a value for only that particular routine. The value is lost w mark). Both routines can use the <u>same variable names</u> to traverse the array <u>without aff</u> (2 marks for showing understanding of underlined words, 1 mark for partial und
11	A For loop repeats a set number of times (1 mark) and the number of times is kn the loop starts (1 mark). A While loop repeats an unknown number of times (1 mark) until a certain cond
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 mar
13	An object (accept variable) called BoardFile is created and filled with data conta
14	Reads a line of the training game file (1 mark), then for each column (1 mark) sp 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 and display a suitable erro

Programming Exercises (Solutions)

Question	Answer
1	***GetRowColumn***

```
def GetRowColumn():
  print
  Column = int(raw_input("Please enter column: "))
  Row = int(raw_input("Please enter row: "))
  while (Row > 9) or (Row < 0) or (Column > 9) or (Column < 0):
    print "Sorry, that is outside the target area.  Please select again."
    Column = int(raw_input("Please enter column: "))
    Row = int(raw_input("Please enter row: "))

  return Row, Column
```



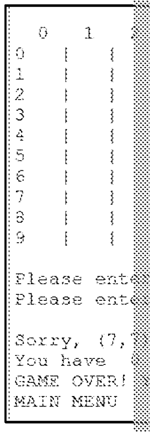
```
Please enter column: 99
Please enter row: 99
Sorry, that is outside the target ar
Please enter column: -66
Please enter row: -66
Sorry, that is outside the target ar
Please enter column: 5
Please enter row: 5

Sorry, (5,5) is a miss.

The board looks like this:

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

Please enter column: 14
Please enter row: -8
Sorry, that is outside the target ar
Please enter column:
```

Question	Answer

2

PlayGame

```
def PlayGame(Board, Ships, Scores):
  GameWon = False
  Torpedoes = 20
  while not GameWon and Torpedoes > 0:
    PrintBoard(Board)
    MakePlayerMove(Board, Ships)
    Torpedoes = Torpedoes - 1
    print "You have ", Torpedoes, " torpedoes left."
    GameWon = CheckWin(Board)
    if GameWon:
      print "All ships sunk!"
      print
    if Torpedoes == 0:
      print "GAME OVER! You ran out of ammo",
```



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

Please ente
Please ente

Sorry, (7,7
You have  (
GAME OVER! 
MAIN MENU
```

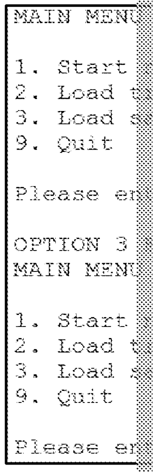
3

DisplayMenu

```
def DisplayMenu():
  print "MAIN MENU",
  print
  print "1. Start new game"
  print "2. Load training game"
  print "3. Load saved game"
  print
```

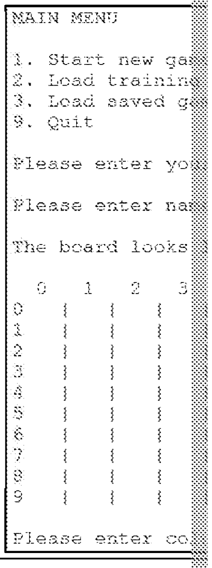
Main program

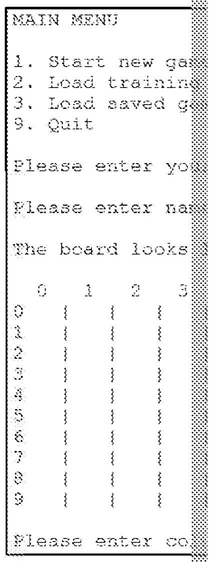
```
if __name__ == "__main__":
  TRAININGGAME = "Training.txt"
  MenuOption = 0
  while not MenuOption == 9:
    Board = SetUpBoard()
    Ships = [["Aircraft Carrier", 5], ["Battleship", 4],
["Submarine", 3], ["Destroyer", 3], ["Patrol Boat", 2]]
    DisplayMenu()
    MenuOption = GetMainMenuChoice()
    if MenuOption == 1:
      PlaceRandomShips(Board, Ships)
      PlayGame(Board,Ships)
    if MenuOption == 2:
      LoadGame(TRAININGGAME, Board)
      PlayGame(Board, Ships)
    if MenuOption == 3:
      print("OPTION 3 EXECUTED")
```



```
MAIN MENU

1. Start 
2. Load t
3. Load s
9. Quit

Please en

OPTION 3 
MAIN MENU

1. Start 
2. Load t
3. Load s
9. Quit

Please en
```

Question	Answer

4	***Main program***
	if MenuOption == 9:
	  sure = raw_input("Are you sure (Y/N)?")
	  if sure != "Y":
	    MenuOption = 0
	
	MAIN MENU
	1. Start new ga
	2. Load trainin
	3. Load saved g
	9. Quit
	Please enter yo
	Please enter na
	The board looks
	  0 1 2 3
	0 | | |
	1 | | |
	2 | | |
	3 | | |
	4 | | |
	5 | | |
	6 | | |
	7 | | |
	8 | | |
	9 | | |
	Please enter co

5	***Main program*** if MenuOption == 3: FileName = raw_input("Please enter name of file: ") LoadGame(FileName, Board) PlayGame(Board, Ships)  MAIN MENU 1. Start new ga 2. Load trainin 3. Load saved g 9. Quit Please enter yo Please enter na The board looks 0 1 2 3 0 | | | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | | 6 | | | 7 | | | 8 | | | 9 | | | Please enter co

Question	Answer

6

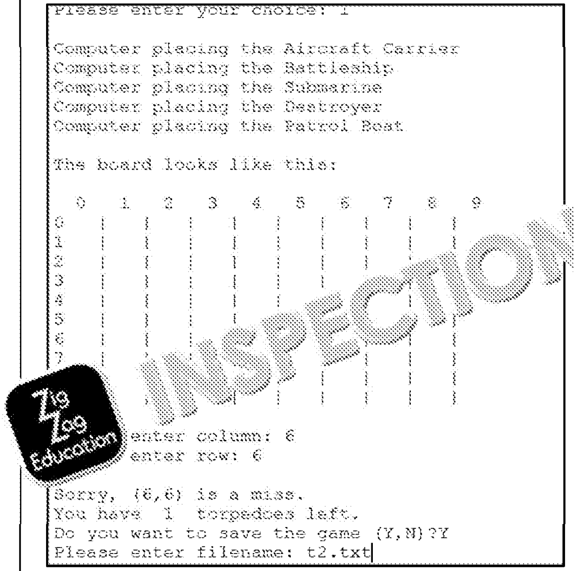
SaveGame

```
def SaveGame(Filename, Board):
  BoardFile = open(Filename, "w")
  for Row in range(10):
    Line = ""
    for Column in range(10):
      Line = Line + Board[Row][Column]
    Line = Line [illegible]
    BoardFile.write(Line)
  BoardFile.close()
```

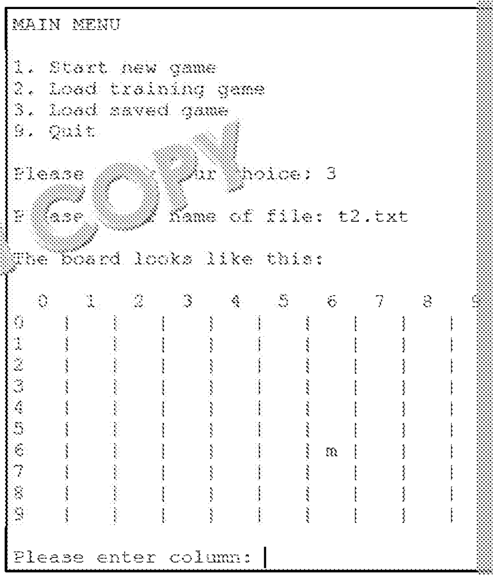
7

PlayGame

```
    Save = raw_input("Do you want to save the game (Y,N)?")
    if Save == "Y":
      Filename = raw_input("Please enter filename: ")
      SaveGame(Filename, Board)
```



```
Please enter your choice: 1

Computer placing the Aircraft Carrier
Computer placing the Battleship
Computer placing the Submarine
Computer placing the Destroyer
Computer placing the Patrol Boat

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: 6

Sorry, (6,6) is a miss.
You have  1  torpedoes left.
Do you want to save the game (Y,N)?Y
Please enter filename: t2.txt
```



```
MAIN MENU

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

Please enter your choice: 3

Please enter Name of file: t2.txt

The board looks like this:

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

Please enter column: 
```

Question	Answer
8	

DisplayMenu

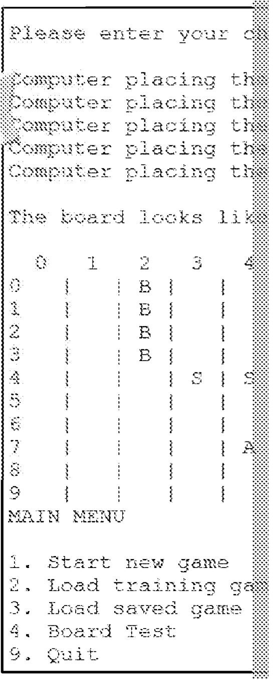
```
print "4. Board Test"
```

Main program

```
if MenuOption == 4:
  PlaceRandomShips(Board, Ships)
  RealBoard(Board)
```

RealBoard

```
def RealBoard(Board):

 print "The board looks like this: "
 print
 print "",
 for Column in range(10):
  print " " + str(Column) + " ",
 print
 for Row in range(10):
  print str(Row),
  for Column in range(10):
   if Board[Row][Column] == "-":
    print " ",
   #elif Board[Row][Column] in ["A","B","S","D","P"]:
   #  print " ",
   else:
    print Board[Row][Column],
   if Column != 9:
    print "|",
  print
```



```
Please enter your c

Computer placing th
Computer placing th
Computer placing th
Computer placing th
Computer placing th

The board looks lik

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

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

Question	Answer
9	

Main program

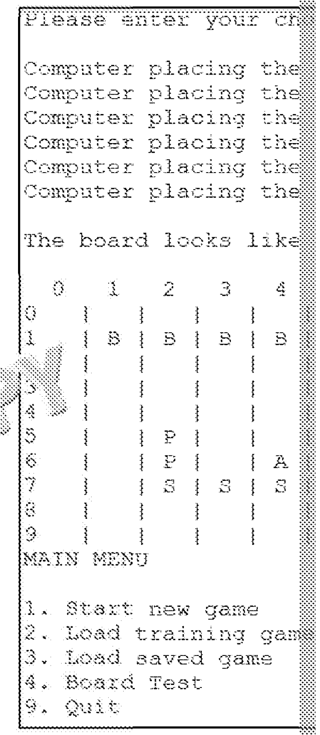
```
if __name__ == "__main__":
  TRAININGGAME = "Training.txt"
  MenuOption = 0
  Scores = [["George",17],["Paul",19],["John",23],["Ri[illegible]",[illegible]],["[illegible]an",30]]
  while not MenuOption == 9:
    Board = SetUpBoard()
    Ships = [["Aircraft Carrier", 5], ["Battleship", 4], ["Submarine", 3], ["Destroyer", 3], ["Patrol
    DisplayMenu()
```

CheckWin

```
def CheckWin(Board):
  for Row in range(10):
    for Column in range(10):
      if Board[Row][Column] in ["A","B","S","D","P","F"]:
        return False
  return True
```

PrintBoard

```
  ...
 if Board[Row][Column] == "-":
       print " ",
     elif Board[Row][Column] in ["A","B","S","D","P","F"]:
       print " ",
```



```
Please enter your ch

Computer placing the
Computer placing the
Computer placing the
Computer placing the
Computer placing the
Computer placing the

The board looks like

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

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

Question

Answer

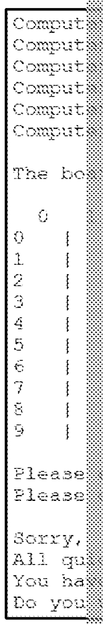
10

MakePlayerMove

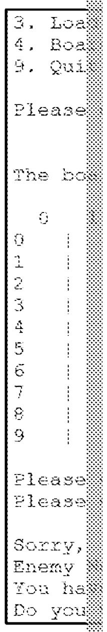
```
 ...
 if Board[Row][Column] == "m" or Board[Row][Column] == "h":
   print "Sorry, you have already shot at the square (" + str(Column) + ","
   + str(Row) + "). Please try again."
 elif Board[Row][Column] == "-":
   print "Sorry, (" + str(Column) + "," + str(Row) + ") is a miss."
   Board[Row][Column] = "m"
   if RadarScan(Board,Row,Column) == True:
     print "Enemy Near!"
   else:
     print "All quiet"
 else:
   ...
```

RadarScan

```
def RadarScan(Board, Row, Column):
  for RowScan in range(Row-1,Row+1):
    for ColumnScan in range(Column-1,Column+1):
      if (Board[RowScan][ColumnScan] != "-" and
      Board[RowScan][ColumnScan] != "h" and
      Board[RowScan][ColumnScan] != "m"):
        return True
  return False
```



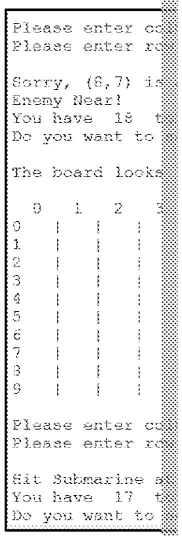
```
Comput
Comput
Comput
Comput
Comput
Comput

The bo

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

Please
Please

Sorry,
All qu
You ha
Do you
```



```
3. Loa
4. Boa
9. Qui

Please

The bo

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

Please
Please

Sorry,
Enemy
You ha
Do you
```

Question	Answer
11	***MakePlayerMove***

```
else:
  ShipName = Board[Row][Column]
  if ShipName == "A":
   ShipName = "Aircraft Carrier"
  elif ShipName == "B":
   ShipName = "Battleship"
  elif ShipName == "D":
   ShipName = "Destroyer"
  elif ShipName == "S":
   ShipName = "Submarine"
  elif ShipName == "P":
   ShipName = "Patrol Boat"
  print "Hit " + ShipName + " at (" + str(Column) + "," + str(Row) + ")."
  Board[Row][Column] = "h"
```



```
Please enter co
Please enter ro

Sorry, (8,7) is
Enemy Near!
You have  18  t
Do you want to 

The board looks

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

Please enter co
Please enter ro

Hit Submarine s
You have  17  t
Do you want to 
```

Question	Answer
12	***PlaceShip***

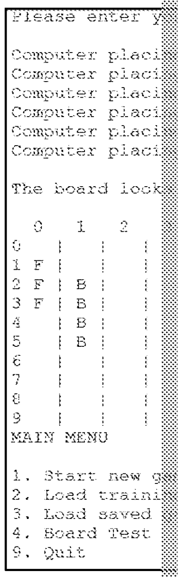
```
 else:
  for Scan in range(Ship[1]):
   Board[Row + Scan][Column + Scan] = Ship[0][0]
```

PlaceRandomShips

```
HorV = random.randint(0, 2)
if HorV == 0:
 Orientation = "v"
elif HorV == 1:
 Orientation = "d"
else:
 Orientation = "h"
```

ValidateBoatPosition

```
def ValidateBoatPosition(Board, Ship, Row, Column, Orientation):
 if (Orientation == "v" or Orientation == "d") and Row + Ship[1] > 10:
  return False
 elif (Orientation == "h" or Orientation == "d") and Column + Ship[1] > 10:
  return False
 else:
  if Orientation == "v":
```



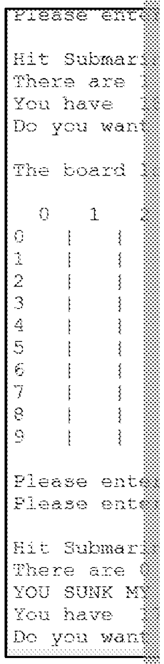
```
Please enter y

Computer placi
Computer placi
Computer placi
Computer placi
Computer placi
Computer placi

The board look

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

1. Start new g
2. Load traini
3. Load saved 
4. Board Test
9. Quit
```

```
   for Scan in range(Ship[1]):
     if Board[Row + Scan][Column] != "-":
       return False
 elif Orientation == "h":
   for Scan in range(Ship[1]):
     if Board[Row][Column + Scan] != "-":
       return False
 else:
   for Scan in range(Ship[1]):
     if Board[Row + Scan][Column + Scan] != "-":
       return False
 return True
```

13

MakePlayerMove

```
ShipName = Board[Row][Column]
ShipLength = -1
for Ship in Ships:
  if Ship[0][0] == ShipName:
    ShipName = Ship[0]
    Ship[1] = Ship[1] - 1
    ShipLength = Ship[1]
print "Hit " + ShipName + " at (" + str(Column) + "," + str(Row) + ")."
print "There are " + str(ShipLength) + " pieces of " + ShipName + " left."
if ShipLength == 0:
  print "YOU SUNK MY " + ShipName.upper()
Board[Row][Column] = "h"
```



```
Please ente

Hit Submari
There are 1
You have  1
Do you want

The board i

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

Please ente
Please ente

Hit Submari
There are 0
YOU SUNK MY
You have  1
Do you want
```

Question	Answer
14	

DisplayMenu

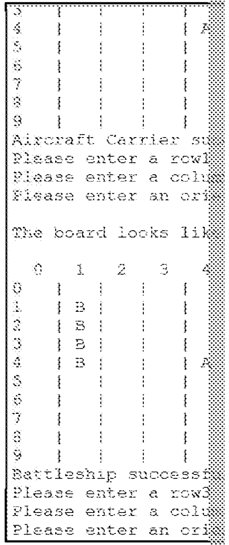
```
  print "5. Manually place ships"
```

Main program

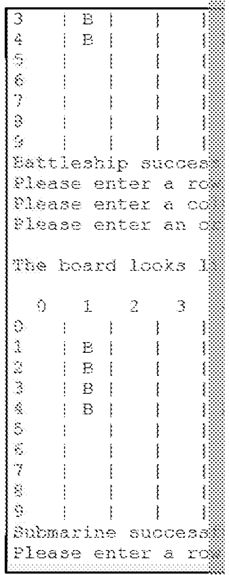
```
  if MenuOption == 5:
    PlaceManualShips(Board, Ships)
    PlayGame(Board,Ships)
```

PlaceManualShips

```
def PlaceManualShips(Board, Ships):
  for Ship in Ships:
    Valid = False
  while not Valid:
    Row = int(raw_input("Please enter a row"))
    Column = int(raw_input("Please enter a column"))
    Orientation = raw_input("Please enter an orientation (v/h)")
    Valid = ValidateBoatPosition(Board, Ship, Row, Column, Orientation)
    if not Valid:
      print "Invalid location.  Please choose again."
    else:
      PlaceShip(Board, Ship, Row, Column, Orientation)
      RealBoard(Board)
      print Ship[0] + " successfully placed at (" + str(Row) +",
      "+ str(Column) +")"
```



```
3   |   |   |   |
4   |   |   |   | A
5   |   |   |   |
6   |   |   |   |
7   |   |   |   |
8   |   |   |   |
9   |   |   |   |
Aircraft Carrier su
Please enter a row1
Please enter a colu
Please enter an ori

The board looks lik

  0   1   2   3   4
0   |   |   |   |
1   | B |   |   |
2   | B |   |   |
3   | B |   |   |
4   | B |   |   | A
5   |   |   |   |
6   |   |   |   |
7   |   |   |   |
8   |   |   |   |
9   |   |   |   |
Battleship successf
Please enter a row3
Please enter a colu
Please enter an ori
```



```
3   | B |   |   |
4   | B |   |   |
5   |   |   |   |
6   |   |   |   |
7   |   |   |   |
8   |   |   |   |
9   |   |   |   |
Battleship succes
Please enter a ro
Please enter a co
Please enter an o

The board looks l

  0   1   2   3
0   |   |   |   |
1   | B |   |   |
2   | B |   |   |
3   | B |   |   |
4   | B |   |   |
5   |   |   |   |
6   |   |   |   |
7   |   |   |   |
8   |   |   |   |
9   |   |   |   |
Submarine success
Please enter a ro
```

Question	Answer
15	

Main program

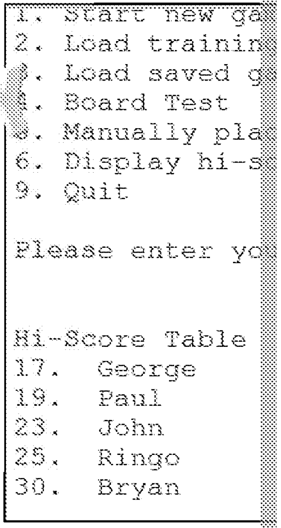
```
if __name__ == "__main__":
  TRAININGGAME = "Training.txt"
  MenuOption = 0
  Scores = [["George",17],["Paul",19],["John",23],["Ringo",25],["Bryan",30]]
  while not MenuOption == 9:
    Board = SetUpBoard()
    Ships = [["Aircraft Carrier", 5], ["Battleship", 4], ["Submarine", 3], ["Destroyer", 3], ["Patrol
    DisplayMenu()
    MenuOption = GetMainMenuChoice()
    if MenuOption == 1:
      PlaceRandomShips(Board, Ships)
      PlayGame(Board,Ships, Scores)
    if MenuOption == 2:
      LoadGame(TRAININGGAME, Board)
      PlayGame(Board, Ships, Scores)
    if MenuOption == 6:
      DisplayHiScores(Scores)
```

DisplayMenu

```
        print "6. Display hi-score table"
```

BubSortScores

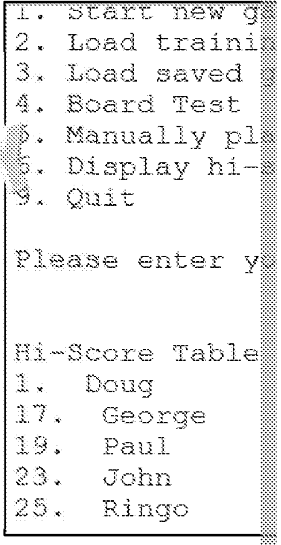
```
def BubSortScores(Scores):
  Changed = True
  while Changed:
    Changed = False
    for i in range(len(Scores)-1):
      if Scores[i][1] > Scores[i+1][1]:
        Changed = True
        Temp = Scores[i+1]
        Scores[i+1] = Scores[i]
        Scores[i] = Temp
```



```
1. Start new ga
2. Load trainin
3. Load saved g
4. Board Test
5. Manually pla
6. Display hi-s
9. Quit

Please enter yo


Hi-Score Table
17.  George
19.  Paul
23.  John
25.  Ringo
30.  Bryan
```

DisplayHiScores

```
def DisplayHiScores(Scores):
  print
  print "Hi-Score Table"
  for Score in Scores:
    print str(Score[1]) +".  " + Score[0]
  print
```



```
1. Start new g
2. Load traini
3. Load saved g
4. Board Test
5. Manually pl
6. Display hi-s
9. Quit

Please enter y


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

PlayGame

```
def PlayGame(Board, Ships, Scores):
  GameWon = False
  Score = 0
  Torpedoes = 20
  while not GameWon and Torpedoes > 0:
    PrintBoard(Board)
    MakePlayerMove(Board, Ships)
    Torpedoes = Torpedoes - 1
    Score = Score + 1
    print "You have ", Torpedoes, " torpedoes left."
    GameWon = CheckWin(Board)
    if GameWon:
      print "All ships sunk!"
      print
      if Score < Scores[-1][1]:
        print "You have a new Hi-Score"
        Scores[-1][1] = Score
        Scores[-1][0] = raw_input("Please enter your name: ")
        BubSortScores(Scores)
    if Torpedoes == 0:
      print "GAME OVER! You ran out of ammo",
      Save = raw_input("Do you want to save the game (Y,N)?")
      if Save == "Y":
        Filename = raw_input("Please enter filename: ")
        SaveGame(Filename, Board)
```

Ideas for modifications	How to im

Electronic Answer Document (EAD) Printout

Name	

ZigZag Education supporting

AS AQA Computer Science Paper 1

Summer 2016: AQA WARSHIPS

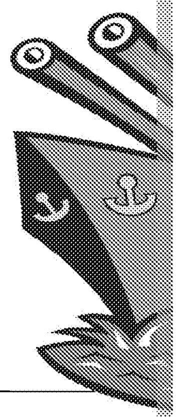
Electronic Answer Document (EAD)

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 Question

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	