3 A programmer is writing a treasure island game to be played on the computer. The island is a rectangular grid, 30 squares by 10 squares. Each square of the island is represented by an element in a 2D array. The top left square of the island is represented by the array element [0, 0]. There are 30 squares across and 10 squares down.

The computer will:

- generate three random locations where treasure will be buried
- prompt the player for the location of one square where the player chooses to dig
- display the contents of the array by outputting for each square:
  - `'.'` for only sand in this square
  - `'T'` for treasure still hidden in sand
  - `'X'` for a hole dug where treasure was found
  - `'O'` for a hole dug where no treasure was found.

Here is an example display after the player has chosen to dig at location [9, 3]:

```
..............................
..............................
..............................
..............................
..............................
........T.....................
..............................
..............................
.........T....................
...X..........................
```

The game is to be implemented using object-oriented programming.

The programmer has designed the class `IslandClass`. The identifier table for this class is:

Identifier	**Data type**	**Description**
`Grid`	`ARRAY[0 : 9, 0 : 29] OF CHAR`	2D array to represent the squares of the island
`Constructor()`		instantiates an object of class `IslandClass` and initialises all squares to sand
`HideTreasure()`		generates a pair of random numbers used as the grid location of treasure and marks the square with `'T'`
`DigHole(Row, Column)`		takes as parameters a valid grid location and marks the square with `'X'` or `'O'` as appropriate
`GetSquare(Row, Column)`	`CHAR`	takes as parameter a valid grid location and returns the grid value for that square from the `IslandClass` object

(a) The programmer designed the pseudocode for the main program as follows:

```
DECLARE Island : IslandClass.Constructor()      // instantiate object

CALL DisplayGrid()                             // output island squares

FOR Treasure ← 1 TO 3                             // hide 3 treasures

   CALL Island.HideTreasure()

ENDFOR

CALL StartDig()                        // user to input location of dig

CALL DisplayGrid()                             // output island squares
```

Write **program code** to implement this pseudocode.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

....[3]

(b) Write **program code** to declare the `IslandClass` and write the constructor method.

The value to represent sand should be declared as a constant.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(c) The procedure `DisplayGrid` shows the current grid data. `DisplayGrid` makes use of the getter method `GetSquare` of the `Island` class.

An example output is:

```
..............................
..............................
..............................
..............................
..............................
........T.....................
..............................
..............................
.........T....................
...X..........................
```

(i) Write **program code** for the `GetSquare(Row, Column)` getter method.

[2]

(ii) Write **program code** for the `DisplayGrid` procedure.

[4]

(d) Write **program code** for the `HideTreasure` method. Your method should check that the random location generated does not already contain treasure.

The value to represent treasure should be declared as a constant.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(e) **(i)** The `DigHole` method takes two integers as parameters. These parameters form a valid grid location. The location is marked with `'X'` or `'O'` as appropriate.

Write **program code** for the `DigHole` method. The values to represent treasure, found treasure and hole should be declared as constants.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

....[3]

(ii) The `StartDig` procedure:

- prompts the player for a location to dig
- validates the user input
- calls the `DigHole` method from **part (e)(i)**.

Write **program code** for the `StartDig` procedure. Ensure that the user input is fully validated.

Programming language used ....

Program code ....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(f) **(i)** The squares in the `IslandClass` grid could have been declared as objects of a `Square` class.

State the term used to describe the relationship between `IslandClass` and `Square`.

....

....[1]

(ii) Draw the appropriate diagram to represent this relationship. Do not list the attributes and methods of the classes.

[2]