CSD 1133 – 2023S

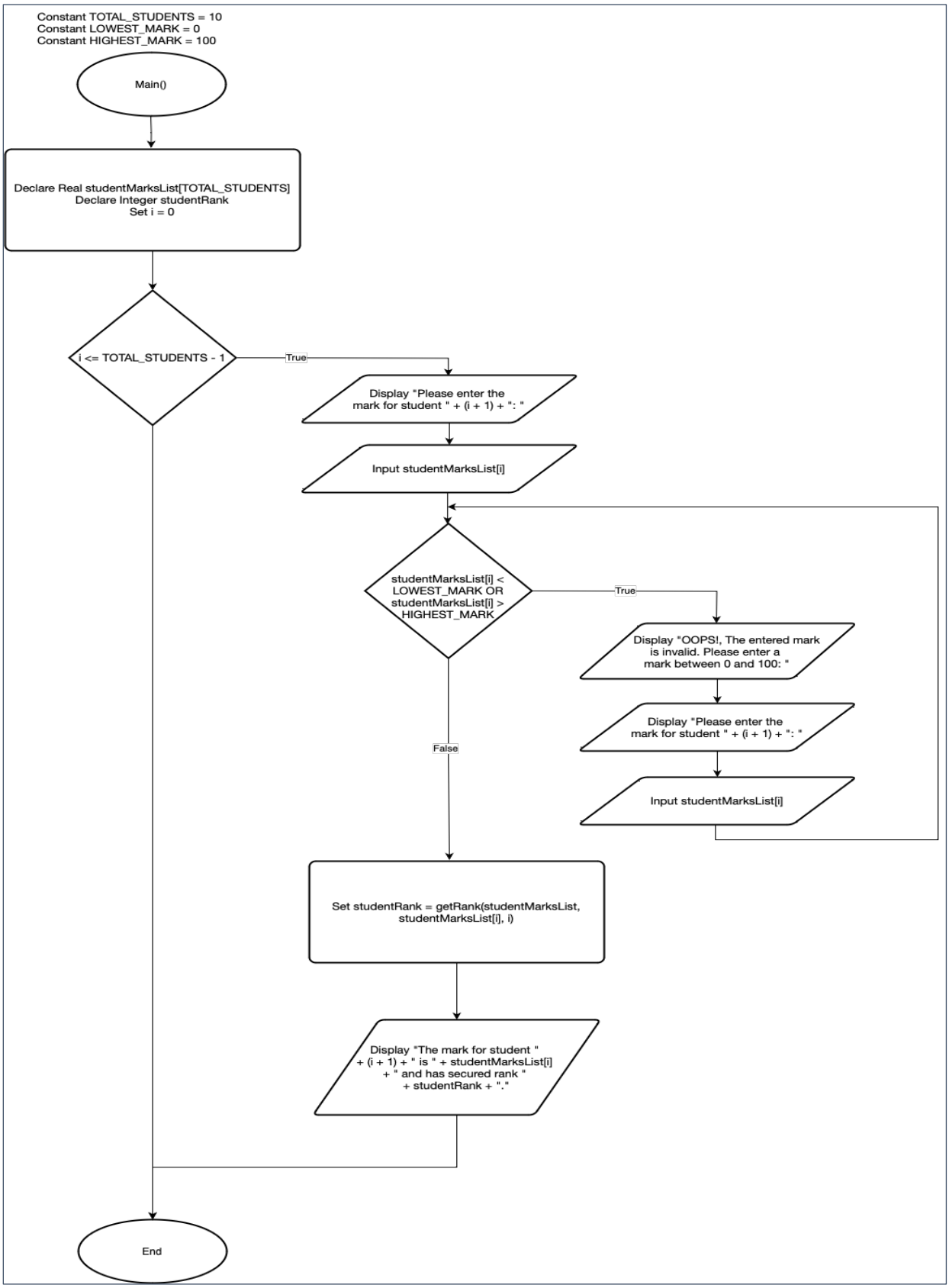
Student ID: 901142
Student Name: Roshan Shrestha
Assignment # 8

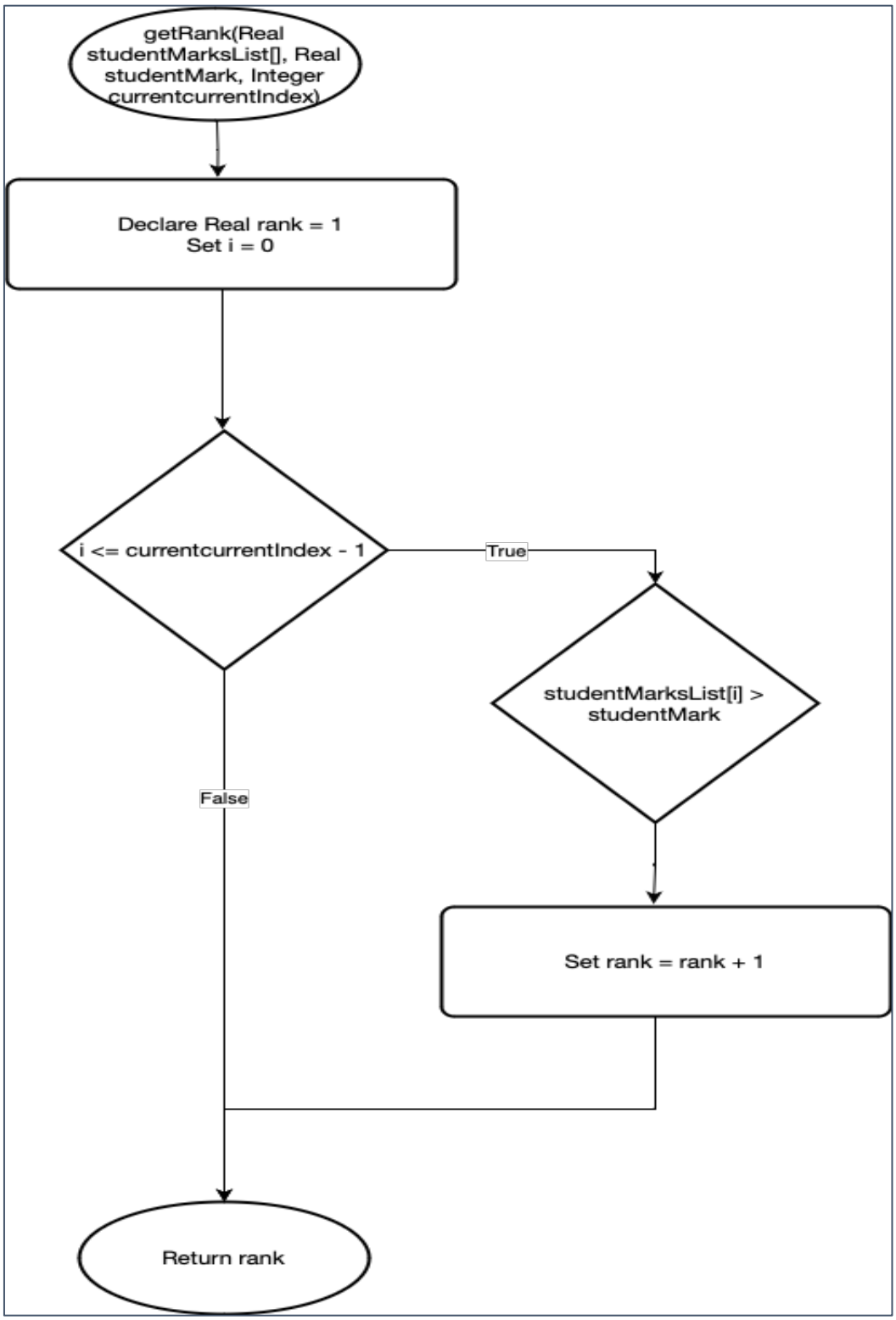
Pseudocode:

```
// Constants to declare the size of the array that holds students' marks in the list
Constant TOTAL_STUDENTS = 10
// Constants for the starting and ending values for the marks range
Constant LOWEST_MARK = 0
Constant HIGHEST_MARK = 100

// Main module, the entry point for the program to execute
Module Main()
   // Declare an array to store the marks of students
   Declare Real studentMarksList[TOTAL_STUDENTS]
   // Declare a variable to store the rank of the student
   Declare Integer studentRank

   // Get the marks from 10 students using a loop
   For i = 0 to TOTAL_STUDENTS - 1
      // Prompt the user to enter the mark for the current student
      Display "Please enter the mark for student " + (i + 1) + ": "
      // Store the input mark in the array
      Input studentMarksList[i]

      // Validate the entered marks to ensure they are within the specified range
      While studentMarksList[i] < LOWEST_MARK OR studentMarksList[i] > HIGHEST_MARK
         Display "OOPS!, The entered mark is invalid. Please enter a mark between 0 and 100: "
         Display "Please enter the mark for student " + (i + 1) + ": "
         Input studentMarksList[i]
      End While

      // Get the rank of the student based on the mark using the getRank function
     Set studentRank = getRank(studentMarksList, studentMarksList[i], i)

      // Display the mark and the rank for the current student
      Display "The mark for student " + (i + 1) + " is " + studentMarksList[i] + " and has secured
rank " + studentRank + "."
   End For
End Module
```

```
// Function to calculate the rank of a student based on their mark
Function Integer getRank(Real studentMarksList[], Real studentMark, Integer
currentcurrentIndex)
   // Declare a variable to store the rank with an initial value of 1
   Declare Real rank = 1

   // Iterate through the list of marks entered for the students
   For i = 0 to currentcurrentIndex - 1
      // Compare the current student's mark with the marks of previous students
      If studentMarksList[i] > studentMark Then
         // If a previous student has a higher mark, increment the rank
         Set rank = rank + 1
      End If
   End For

   // Return the calculated rank for the current student
   Return rank
End Function
```

Flow Chart:


Constant TOTAL_STUDENTS = 10
Constant LOWEST_MARK = 0
Constant HIGHEST_MARK = 100
Main()
Declare Real studentMarksList[TOTAL_STUDENTS]
Declare Integer studentRank
Set i = 0
i <= TOTAL_STUDENTS - 1
True
Display "Please enter the mark for student " + (i + 1) + ": "
Input studentMarksList[i]
studentMarksList[i] < LOWEST_MARK OR studentMarksList[i] > HIGHEST_MARK
True
Display "OOPS!, The entered mark is invalid. Please enter a mark between 0 and 100: "
Display "Please enter the mark for student " + (i + 1) + ": "
Input studentMarksList[i]
False
Set studentRank = getRank(studentMarksList, studentMarksList[i], i)
Display "The mark for student " + (i + 1) + " is " + studentMarksList[i] + " and has secured rank " + studentRank + "."
End


getRank(Real studentMarksList[], Real studentMark, Integer currentcurrentIndex)
Declare Real rank = 1
Set i = 0
i <= currentcurrentIndex - 1
True
False
studentMarksList[i] > studentMark
Set rank = rank + 1
Return rank