Workbook O Level & IGCSE Computer Science

Solution of Pre-release Material

for Summer 2017

Inqilab Ruknuddin Patel

Topical Past Paper Questions

Lecture Notes

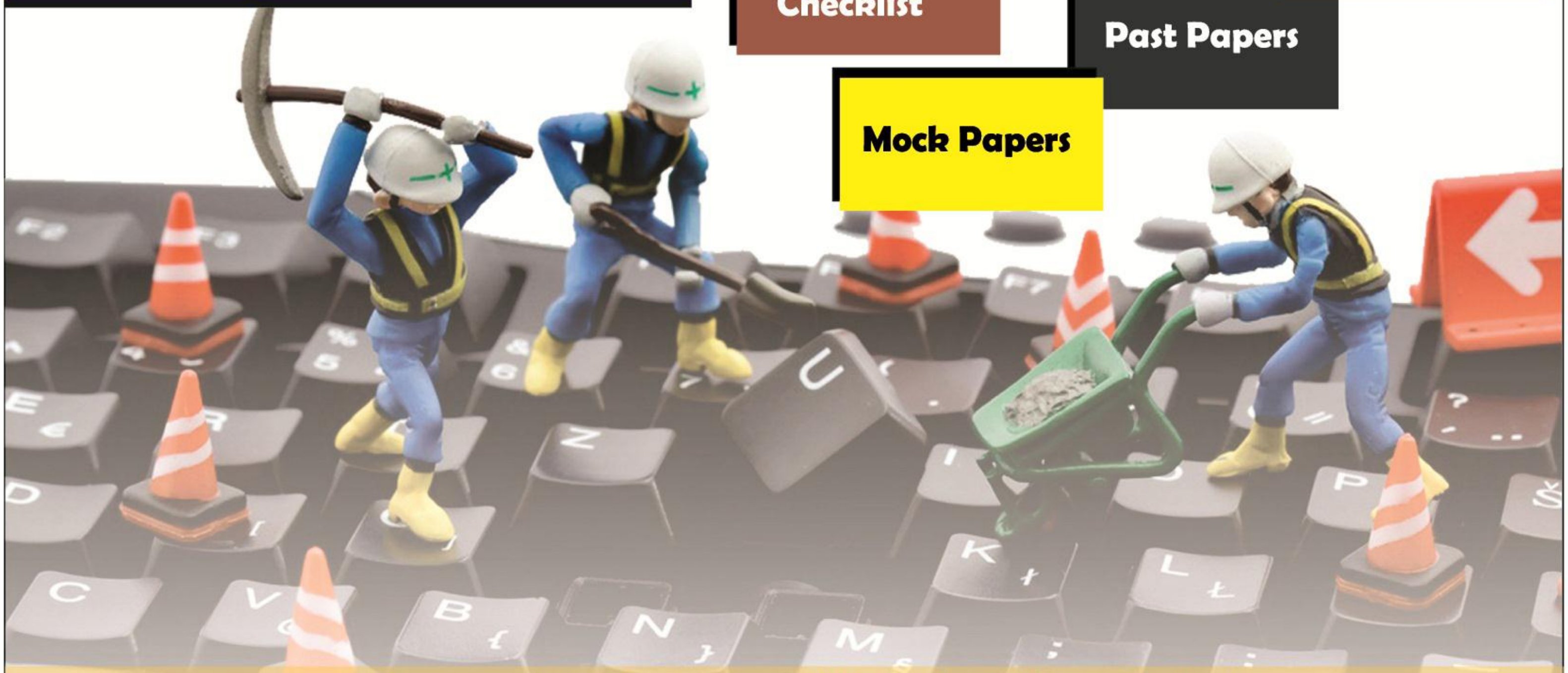
Practice Questions

Revision Guide

Revision Checklist

Past Papers

Mock Papers



Computer Science With Inqilab Patel

+923002724734 /inqilabpatel inqilab patel

inqilab inqilab-patel ruknuddin.com

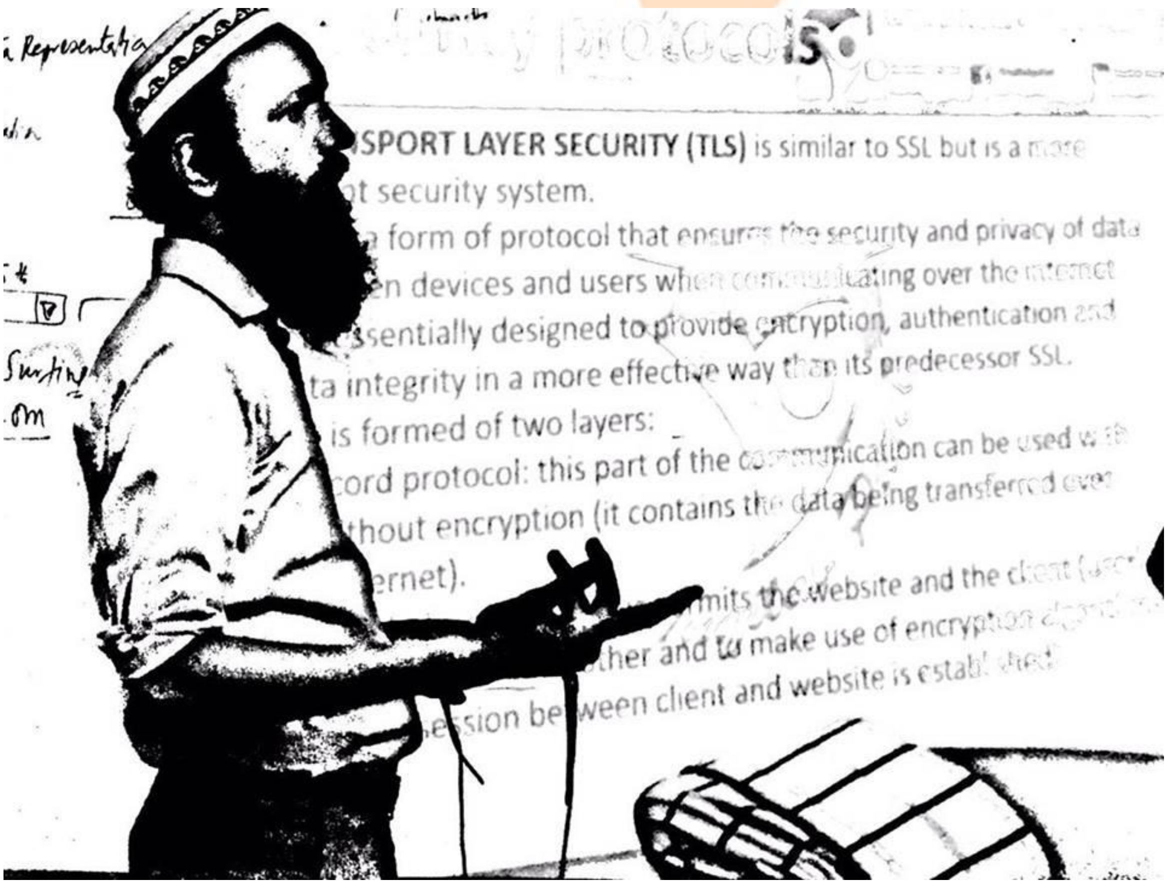
About the developer of this solution

Inqilab Patel is an O &A Level Computer Teacher. He has taught in many schools including Yaqeen Model School, Karachi Cadet School, KN Academy, Beacon House and The City School, **PAF Chapter** and **Nakhlah** Boys Campus Society. **Cambridge** has selected him as a **Member** of **Cambridge Editorial Review Board**. He is also associated with **Aga Khan University Examination Board**, **Karachi Board of Secondary Education** and **Sindh Board of Technical Education**.

His entire career path revolves around computer science; either he was a student or a teacher. He got a chance to polish his skills of teaching and studying more about computers at various levels which has given him great confidence in presenting himself for any senior level position of transferring his knowledge to the youth.

Inqilab Patel knows a lot of methods of teaching computers and has developed tutorial notes, worksheets and assignments for my students. He also maintains a website (www.ruknuddin.com) which is specifically designed for the support of those who want to excel in IGCSE computer science. He also regularly contributes material to CIE teacher support website, for which he receives appreciation from different people across the world.

He has also received various training in innovative and special methods of teaching this subject.



Paper 1 Theory

This is a compulsory question paper, consisting of short-answer and structured questions set on Section 1 of the syllabus content. All questions are compulsory. Candidates will answer on the question paper.

Paper 2 Problem-solving and Programming

This is a compulsory question paper, consisting of short-answer and structured questions set on Section 2 of the syllabus content. All questions are compulsory. Candidates will answer on the question paper.

20 of the marks in this paper are from questions set on tasks provided in the Paper 2 Problem-solving and Programming pre-release material.

Centres need to be aware that in order to prepare best their candidates for this paper, they should plan for sufficient practical sessions within their lesson timetable and teach the contents of the section in a largely practical way. Candidates will be expected to be able to program in a high-level programming language to be chosen by the Centre. The programming language should be procedural.

There will be some examining of knowledge with understanding, but most of the credit will be for using techniques and skills to solve problems. The examination questions will require candidates to have practical programming experience, including writing their own programs, executing (running), testing and debugging them. Knowledge of programming language syntax will not be examined; in all cases the logic will be more important than the syntax.

Paper 2 Problem-solving and Programming pre-release material

The Paper 2 Problem-solving and Programming pre-release material will be made available to Centres the January before the June examination, and the July before the November examination. It will also be reproduced in the question paper. Candidates are not permitted to bring any prepared material into the examination.

Centres are advised to encourage their candidates to develop solutions to tasks using a high-level programming language, such as Visual Basic, Pascal/Delphi or Python. The purpose of the pre-release material tasks is to direct candidates to some of the topics which will be examined in Paper 2. Teachers are expected to incorporate these tasks into their lessons and give support in finding methods and reaching solutions. 20 of the marks in this paper will be from questions testing candidates' understanding gained from developing programmed solutions to these tasks.

Here is a copy of pre-release material

In Preparation for the examination candidates should attempt the following practical tasks by **writing and testing a program or programs.**
The Organizer of a senior citizens' club has arranges outings for the members. For each of these outings a coach is hired, meals at a restaurant are reserved and tickets for the theatre are booked. A program is required to work out the costs and provide a printed list showing everyone on the outing.
Write and test a program for the club organizer.

- Your program must include appropriate prompts for the entry of data.
- Error message and other output need to be set out clearly.
- All variables, constants and other identifiers must have meaningful names.

You will need to complete these three tasks. Each task must be fully tested.

TASK 1 - Work out the total cost of the outing.
The organizer finds out how many senior citizens are interested in the outing. The program for TASK 1 works out the cost for the information.

Number of people	Hire of couch ($)	Cost of meal ($)	Cost of theatre ticket ($)
12-16	150	14.00	21.00
17-26	190	13.50	20.00
27-39	225	13.00	19.00

The minimum number of senior citizens needed for the outing to go ahead is 10; there cannot be more than 36 senior citizens on the outing. A minimum of two carers go on the outing. With an additional carer needed if more than 24 citizens go on the outing. Carers do not have to pay anything for the outing. Workout the total cost per person for the senior citizens.

TASK 2 - Record who is going on the outing and how much has been paid.
Using your results from Task 1, record the names of the people on the outing and the amount they have paid; include the carers on the outing. If there are spare places on the coach then extra people can be added; they are charged the same price as the other citizens. Calculate the total amount of money collected. Print out the list of the people on the outing.

TASK 3 - Identify the break-even point or profit that will be made on the outing.
Show whether the outing has made a profit or has broken even using the estimated cost from the TASK 1 and money collected from TASK 2.

To understand the workings in tasks firstly fill in the following trace table. Use given formulae and following test data:

5, 10, 12, 14, 20, 24, 30, 33, 36, 40

Senior Citizens	Carer	People	Coach	Per Person Meal	Per Person Ticket	Estimated Cost	Per Person Cost	Extra People	Extra Total	Amount Collected	Extra Expense	Total Expenses	Profit	OUTPUT
(Input)	(Selection)	(S_Citizen+Care)	Selection			Coach+(PP_Meal+PP_Ticket) * People)	(Estimated_Cost/S_Citizen)	(Selection)	(Extra_People * PP_Cost)	(Estimated_Cost + Extra_Amount)	(PP_Meal + PP_Ticket) * Extra_People)	(Estimated_Cost + Extra_Expense)	Amount Collected – Total Expenses	

The solved trace table

Senior Citizens	Carer	People	Coach	Per Person Meal	Per Person Ticket	Estimated Cost	Per Person Cost	Extra People	Extra Total	Amount Collected	Extra Expense	Total Expenses	Profit	OUTPUT
(Input)	(Selection)	(S_Citizen+Care)	Selection			Coach+(PP_Meal+PP_Ticket) * Peaple)	(Estimated_Cost/S_Citizen)	(Selection)	(Extra_People * PP_Cost)	(Estimated_Cost + Extra_Amount)	(PP_Meal + PP_Ticket) * Extra_People)	(Estimated_Cost + Extra_Expense)	Amount Collected – Total Expenses	
5														Error: out of range
10	2	12	150	14	21	570	47.5	4	190.0	760.0	140	710	50.0	Outing has generated a profit
12	2	14	150	14	21	640	45.71429	2	91.4	731.4	70	710	21.4	Outing has generated a profit
14	2	16	150	14	21	710	44.375	0	0.0	710.0	0	710	0.0	Outing has broken even
20	2	22	190	13.5	20	927	42.13636	4	168.5	1095.5	134	1061	34.5	Outing has generated a profit
24	2	26	190	13.5	20	1061	40.80769	0	0.0	1061.0	0	1061	0.0	Outing has broken even
30	3	33	225	13	19	1281	38.81818	6	232.9	1513.9	192	1473	40.9	Outing has generated a profit
33	3	36	225	13	19	1377	38.25	3	114.8	1491.8	96	1473	18.8	Outing has generated a profit
36	3	39	225	13	19	1473	37.76923	0	0.0	1473.0	0	1473	0.0	Outing has broken even
40														Error: out of range

TASK 1 - Work out the total cost of the outing.
The organizer finds out how many senior citizens are interested in the outing. The program for TASK 1 works out the cost for the information.

Number of people	Hire of couch ($)	Cost of meal ($)	Cost of theatre ticket ($)
12-16	150	14.00	21.00
17-26	190	13.50	20.00
27-39	225	13.00	19.00

The minimum number of senior citizens needed for the outing to go ahead is 10; there cannot be more than 36 senior citizens on the outing. A minimum of two carers go on the outing. With an additional carer needed if more than 24 citizens go on the outing. Carers do not have to pay anything for the outing. Workout the total cost per person for the senior citizens.

Data structure:

A **data structure** is a specialized format for organizing and storing **data**. General **data structure** types include the array, list, variables, the file, the record, the table, and so on.
Variables: A variable is a memory location. It has a name (an identifier) that is associated with that location. The value associated with a variable name may change during program execution.
Array: A variable that can store multiple data items.
List: a set of data items grouped together.

Variabels:

Data structure name	Data Type	Purpose
S_Citizen	Integer	To input and store number of senior citizens interested in outing
Carer	Integer	To calculate number of accompanied carers
People	Integer	To calculate total people going on outing
Coach	Real	To select hire of coach in $
PP_Meal	Real	To select per person meal in $
PP_Ticket	Real	To select per person theatre ticket in $
Estimated_Cost	Real	To calculate total estmated cost of outing
PP_Cost	Real	To calculate per citizen cost

. Validation:

To reject if out of range number of senior citizens are intereted in outing

```
WHILE S_Citizen<10 or S_Citizen>36 DO
    PRINT "Error: due to out of range senior citizens"
    PRINT "Re-enter a valid number between 10 to 36"
    READ S_Citizen
ENDWHILE
```

Test Data:

To check corectness of pseudo code

Test Data Set	Purpose
12, 22, 32	To check input of Normal Data
10, 30, 36	To check input of Extreme Data
5, 40	To check rejection of Abnormal Data

Formulae:

To add extra carer & total people

```
IF S_Citizen>24 THEN Carer ← Carer + 1
People ← S_Citizen + Carer
```

To calulate estimated costl

```
Estimated_Cost ← Coach + (People * PP_Meal) + (People * PP_Ticket)
```

To calulate per person cost

```
PP_Cost ← Estimated_Cost / S_Citizen
```

Pseudocode

```
// Declaration of variables
    DECLARE S_Citizen, Carer, People: Integer
        DECLARE Coach, PP_Meal, PP_Ticket, Estimated_Cost, PP_Cost: Real

//The organizer finds out how many senior citizens are interested in the outing.
//Input and store number of senior citizens interested in outing

        PRINT "Enter number of senior citizens interested in outing:"
        READ  S_Citizen

// The minimum number of senior citizens needed for the outing to go ahead is 10;
there cannot be more than 36 senior citizens on the outing.

        //Validation of number of senior citizens 10-36
        WHILE S_Citizen<10 OR S_Citizen>36 DO
            PRINT "Error: due to out of range senior citizens"
            PRINT "Re-enter a valid number between 10 to 36"
            READ S_Citizen
        ENDWHILE
```

$\big\}$ Validation of number of senior citizens

// A minimum of two carers go on the outing. With an additional carer needed if more than 24 citizens go on the outing.

```
Care ← 2
IF S_Citizen>24 THEN Carer ← Carer + 1

People ← S_Citizen + Carer
```

// works out the total cost, per citizen cost using given information in the table

```
IF People<=16 THEN
        Coach ← 150
        PP_Meal ← 14
        PP_Ticket ← 21
ELSEIF  People<=26 THEN
        Coach ← 190
        PP_Meal ← 13.5
        PP_Ticket ← 20
ELSE
        Coach ← 225
        PP_Meal ← 13
        PP_Ticket ← 19
ENDIF

Estimated_Cost ← Coach + (People * PP_Meal) + (People * PP_Ticket)
PP_Cost ← Estimated_Cost / S_Citizen
```

Visual Basic Code

```
Module Module1

    Sub Main()
        Dim S_Citizen, Carer, People As Integer
        Dim Coach, PP_Meal, PP_Ticket, Estimated_Cost, PP_Cost As Single


        Console.Write("Enter number of senior citizens interested in outing: ")
        S_Citizen = Console.ReadLine

        While S_Citizen < 10 Or S_Citizen > 36
            Console.WriteLine("Error: due to out of range senior citizens")
            Console.Write("Re-enter a valid number between 10 to 36 ")
            S_Citizen = Console.ReadLine
        End While

        Carer = 2
        If S_Citizen > 24 Then Carer = Carer + 1
        People = S_Citizen + Carer

        If People <= 16 Then
                    Coach = 150
                    PP_Meal = 14
                    PP_Ticket = 21
        ElseIf People <= 26 Then
                    Coach = 190
                    PP_Meal = 13.5
                    PP_Ticket = 20
        Else
                    Coach = 225
                    PP_Meal = 13
                    PP_Ticket = 19
        End If

        Estimated_Cost = Coach + (People * PP_Meal) + (People * PP_Ticket)
        PP_Cost = Estimated_Cost / S_Citizen

        Console.ReadKey()

    End Sub

End Module
```

TASK 2 - Record who is going on the outing and how much has been paid.
Using your results from Task 1, record the names of the people on the outing and the amount they have paid; include the carers on the outing. If there are spare places on the coach then extra people can be added; they are charged the same price as the other citizens. Calculate the total amount of money collected. Print out the list of the people on the outing.
.

Data structure:

A **data structure** is a specialized format for organizing and storing **data**. General **data structure** types include the array, the file, the record, the table, and so on.

Data structure name	Data Type	Purpose
Senior_Name[1:S_Citizen] or Senior_Name[S_Citizen]	String	To input,store and name of senior citizens interetested in outing
Senior_Amount[1:S_Citizen]	Real	To input and store amount paid by senior citizens
Carer_Name[1:Carer]	String	To input,store and name of carers
Extra_Name[1:Extra_People]	String	To input,store and name of extra people going on outing
Extra_Amount[1: Extra_People]	Real	To input,store and amount paid by extra people

Variabels:

Variable Name	Data Type	Purpose
Extra_People	Real	To calculate extra people
Extra_Total	Real	To calculate total collected class total
Amount_Cllected	Real	To calculate Total money collected
Index	Integer	To use for index in array

Formulae:

To caluclate extra total collection from extra people

```
    Extra_Total ← Extra_People * PP_Cost
```

To caluclate total amount collect

```
    Amount_Collected ← Estimated_Cost + Extra_Total
```

Pseudocode of Task 2

//Declaration of variables

```
    DECLARE Senior_Name[1:S_Citizen], Carer_Name[1:Carer]: String
    DECLARE Senior_Amount[1:S_Citizen]: Real
        DECLARE Index: Integer
        DECLARE Amount_Collected, Extra_Total, Extra_People: Real
```

//Record the names of the people on the outing and the amount they have paid; include the carers on the outing.

```
          For Index = 1 To S_Citizen
                    PRINT "Enter name of senior citizen : "
               INPUT Senior_Name[Index]
                    PRINT "Enter amount paid by senior citizen : "
               INPUT Senior_Amount[Index]
        Next Index

          For Index = 1 To Carer
                    PRINT "Enter name of carer : "
               INPUT Carer_Name[Index]
        Next Index
```

// If there are spare places on the coach then extra people can be added; they are charged the same price as the other citizens.

```
        IF People<=16 THEN
                  Extra_People ← 16 - People
        ELSEIF People<=26 THEN
                  Extra_People ← 26 - People
        ELSE
                  Extra_People ← 36 - People
        ENDIF

        DECLARE Extra_Name[1:Extra_People] : String
        DECLARE Extra_Amount[1:Extra_People]:Real

        For Index = 1 To Extra_People
                  PRINT "Enter name of extra person going on outing : "
             INPUT Extra_Name[Index]
                  PRINT "Enter amount paid by extra person : "
             INPUT Extra_Amount[Index]
        Next Index
```

// They are charged the same price as the other citizens.

```
             Extra_Total ← Extra_People * PP_Cost
```

//Calculate the total amount of money collected.

```
      Amount_Collected ← Estimated_Cost + Extra_Total
```

// Print out the list of the people on the outing.

```
      PRINT "List of senior citizens going on outing"
      For Index = 1 To S_Citizen
           PRINT Senior_Name[Index]
      Next Index

      PRINT "List of carers going on outing"
        For Index = 1 To Carer
                PRINT Carer_Name[Index]
      Next Index

      PRINT "List of extra people going on outing"
      For Index = 1 To Extra_People
                PRINT Extra_Name[Index]
      Next Index
```

Visual Basic Code of Task 2

```
Module Module1
    Sub Main()

       Dim Senior_Name(S_Citizen), Carer_Name(Carer) As String
        Dim Senior_Amount(S_Citizen) As Single
        Dim Index As Integer
        Dim Amount_Collected, Extra_Total, Extra_People As Single

        For Index = 1 To S_Citizen
            Console.Write("Enter name of senior citizen : ")
            Senior_Name(Index) = Console.ReadLine
            Console.Write("Enter amount paid by senior citizen : ")
            Senior_Amount(Index) = Console.ReadLine
        Next Index

        For Index = 1 To Carer
            Console.Write("Enter name of carer : ")
            Carer_Name(Index) = Console.ReadLine
        Next Index

        If People <= 16 Then
            Extra_People = 16 - People
        ElseIf People <= 26 Then
            Extra_People = 26 - People
        Else
            Extra_People = 36 - People
        End If
        Dim Extra_Name(Extra_People) As String
        Dim Extra_Amount(Extra_People) As Single

        For Index = 1 To Extra_People
            Console.Write("Enter name of extra person going on outing : ")
            Extra_Name(Index) = Console.ReadLine
            Console.Write("Enter amount paid by extra person : ")
            Extra_Amount(Index) = Console.ReadLine
        Next Index



       Extra_Total = Extra_People * PP_Cost

        Amount_Collected = Estimated_Cost + Extra_Total

        Console.WriteLine("List of senior citizens going on outing")
        For Index = 1 To S_Citizen
            Console.WriteLine(Senior_Name(Index))
        Next Index

        Console.WriteLine("List of carers going on outing")
        For Index = 1 To Carer
            Console.WriteLine(Carer_Name(Index))
        Next Index

        Console.WriteLine("List of extra people going on outing")
        For Index = 1 To Extra_People
            Console.WriteLine(Extra_Name(Index))
        Next Index

        Console.ReadKey()
    End Sub

End Module
```

TASK 3 - Identify the break-even point or profit that will be made on the outing.
Show whether the outing has made a profit or has broken even using the estimated cost from the TASK 1 and money collected from TASK 2.

Variabels:

Variable Name	Data Type	Purpose
Extra_Expense	Real	To calculate extra expense on extra people
Total_Expenses	Real	To calculate total expenses inlcuding extra people
Profit	Real	To calculate profit

Formulae:

To calculate extra expenses on extra people

```
        Extra_expense ← Extra_People *  (PP_Meal + PP_Ticket)
```

To calculate total exenses

```
        Total_Expense ← Extra_expense + Estimated Cost
```

To calculate profit

```
        Profit ← Amount_Collected –  Total_Expenses
```

Pseudocode

```
// Declaration

        DECLARE Extra_Expense, Total_Expense, Profit: Real
// Calculations

        Extra_expense ← Extra_People *  (PP_Meal + PP_Ticket)

        Total_Expense ← Extra_expense + Estimated Cost

        Profit ← Amount_Collected –  Total_Expenses

// outing has made a profit or has broken even

    IF Profit>0 THEN
        PRINT “Outing has generated a profit”
    ELSE
        PRINT “Outing has broken even”
    ENDIF
```

VB Code of Task 3

```
Module Module1
    Sub Main()

        Dim Extra_Expense, Total_Expense, Profit As Single


        Extra_Expense = Extra_People * (PP_Meal + PP_Ticket)

        Total_Expense = Extra_Expense + Estimated_Cost

        Profit = Amount_Collected - Total_Expense


        If Profit > 0 Then
            Console.WriteLine("Outing has generated a profit")
        Else
            Console.WriteLine("Outing has broken even")
        End If


        Console.ReadKey()

    End Sub

End Module
```