

Tutorial 3 MT262

Tutor : Rifat Hamoudi
Staff No. : 00567451
Pager No. : 07669-801 509

I have put this tutorial on the web. This tutorial can be viewed and downloaded from <http://www.users.totalise.co.uk/~rifat> then selecting MT262 Tutorials then Tutorial 4.

1) Design and implement a C++ program that counts the number of letters in a user's name.

An example output is as shown below :



```
strings
Enter your full name followed by space: Rifat Hamoudi
Number of letters in your name is 12
```

Trace the program using Borland C++ Builder provided with this course and put a watch on the loop counter and any other variable.

2) This question is to do with preparation for writing a proper Database Application using Object Oriented C++ :

- (a) Write a function to convert a character to uppercase
- (b) The company want to create a database to keep track of its employees name, address and salary. Write a class with appropriate method definitions any other necessary variables to capture what the company wants. Your methods should be one for initialisation, one for adding an employee and one for displaying the employee details
- (c) Sketch design for files needed to write the database application and what should go in each file

3) Write a database application for the company to add and display its employees. For this tutorial assume the maximum number of employees is 2 and just write data to RAM so don't worry about data to file. Add error check to ensure that user choice is always processed. Also use the class definition in Question 2(b) to record and display the employee details. Test your program to make sure that it works.

Answer to Question 1

The design

Top level design

- 1 read in user name
- 2 count the letters in the user's name
- 3 write out the letters in the user's name

Final design using stepwise refinement

- 1.1 write out "Enter your full name followed by space: "
- 1.2 read in Line
- 2.1.1 Index <- 1
- 2.1.2 LetterCount <- 0
- 2.1.3 PreviousLetter <- ' '
- 2.2 **loop while** Index <= Length(Line)
- 2.3.1 **if** PreviousLetter != ' ' **then**
- 2.3.2 LetterCount <- LetterCount + 1
- 2.3.3 **ifend**
- 2.3.4 PreviousLetter <- Line[Index]
- 2.3.5 Index <- Index + 1
- 2.4 **loopend**
- 3.1 write out "Number of letters in your name is", LetterCount

The C++ source code

```
/*
    Unit      :   MT262
    Tutorial:   Tutorial 3
    Title     :   Count letters in name program with semantic error
    Author    :   Rifat Hamoudi
    Version   :   0.1
    Date      :
*/

#include "MT262io.h" // modules defined in it e.g. ReadInt
#pragma hdrstop //tells Builder how to deal with library files

//-----
// Project|Add to project -> MT262io.lib
USELIB("MT262io.lib");
//-----
-
#pragma argsused // to avoid the production of warning messages
                // relating to argc and **argv

int main(int argc, char **argv)
{
    AnsiString Line;
    int Index;
    int LetterCount;
    char PreviousLetter;

    Line = ReadStringPr("Enter your full name followed by space: ");
    Index = 1;
    LetterCount = 0;
    PreviousLetter = ' ';

    while (Index <= Length(Line))
    {
        if (PreviousLetter != ' ')
            LetterCount = LetterCount + 1;
        PreviousLetter = Line[Index];
        Index = Index + 1;
    }
    WriteIntPrCr("Number of letters in your name is", LetterCount);
    getchar(); // to keep the display on until you press Enter key
    return 0;
}
```

The above code has semantic error, detect the semantic error using the trace facility and correct the program?

How to trace a program in C++ Builder

- 1) Load your code
- 2) Click on Run then Step Over menu option or Click F8
- 3) Keep clicking F8 to trace through the program line by line
- 4) When you want to get into a function implementation click on Run then Trace into or F7, this will take you to the function or method implementation body
- 5) To add a watch highlight the variable and click on Run then Add Watch or Ctrl+F5, a separate window will appear which will contain the variable you want to watch and its current value, as you trace through the program the value changes accordingly

Answer to Question 2

(a)

```
char Tut3_uppercase(char selection)
{
    if ((selection >= 'a') && (selection <= 'z'))
        selection = selection - 32;

    return selection;
}
```

(b)

```
#define MaxEmp 2

class CompanyType
{
private:

    struct EmployeeType
    {
        int Salary;
        AnsiString Address;
        AnsiString FullName;
    };
    EmployeeType Employee[MaxEmp];

public:

    void Init(void);
    AnsiString AddEmp(AnsiString EmpName, AnsiString EmpAddress, int
EmpSalary);
    void DispEmp(void);
};
```

(c)

The files can be divided into 5 files as follows :

Tut3_3.cpp : Main file which contains the main driving code.

Tut3_3_methodimp.cpp : File containing the implementation of the methods in the Company class

Tut3_3_methodimp.h : contains the class definition used by the database

Tut3_3_funcimp.cpp : contains the implementation of functions used by the methods in the Company class and the main driver

Tut3_3_funcimp.h : contains function prototypes (definition)

Answer to Question 3

C++ Builder Project Tut3_3.bpr – Relationships between components

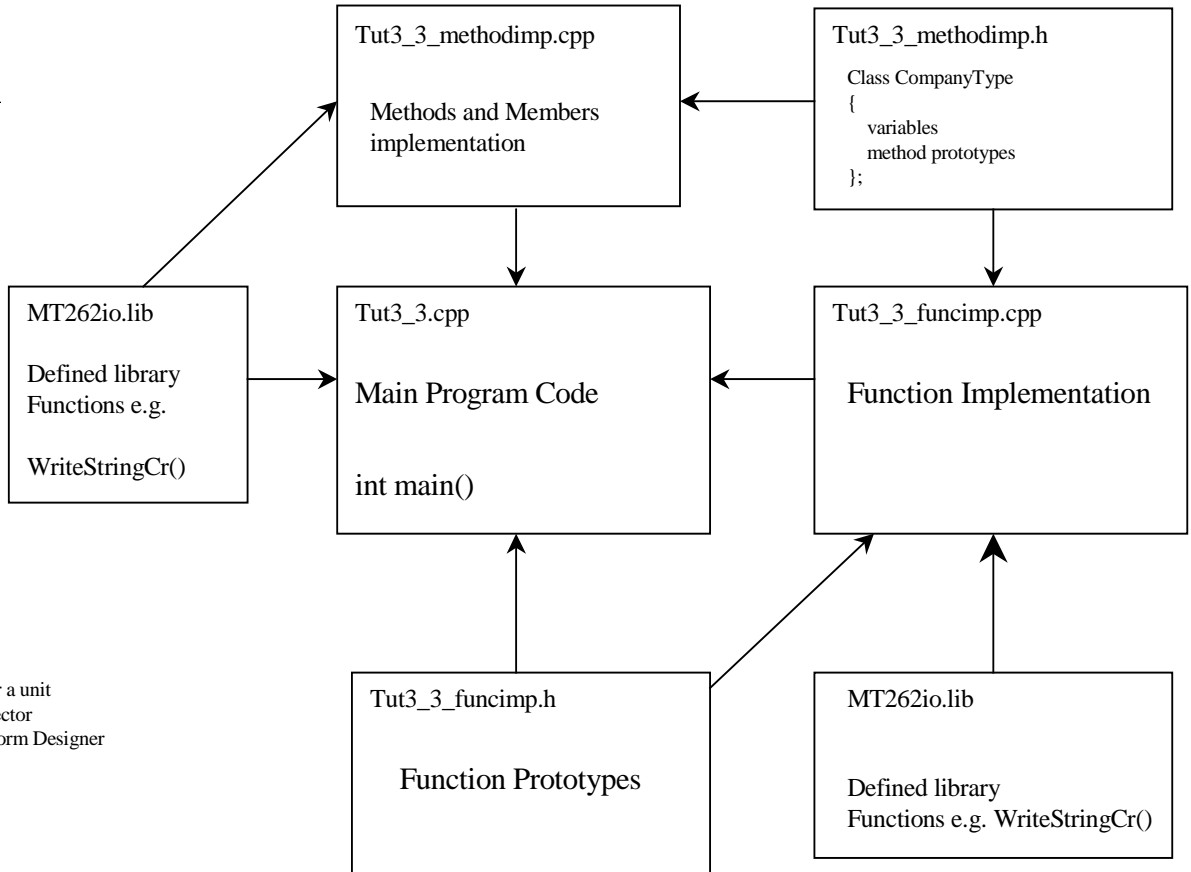
Tut3_3.bpr project

To create a new Windows project:

File!New Application

File!Save Project As

1. Unit
2. Project



To create a new unit:

File!New Unit

Builder keyboard shortcuts:

Ctrl+F6 to view *.hdr file for a unit

F11 to view Object Inspector

F12 to switch between Form Designer
and Code Editor

Tut3_3.cpp

```
/*
    Unit      :   MT262
    Tutorial:   Tutorial 3_3
    Title     :   Company Database Application Program
    Author    :   Rifat Hamoudi
    Version   :   0.1
    Date      :
*/

// Main file for Tutorial 3 database application

#include <condefs.h>
#pragma hdrstop
#include <conio.h>
#include "MT262io.h"
#include "tut3_3_funcimp.h"

//-----
USELIB("Mt262io.lib");
USEUNIT("tut3_3_funcimp.cpp");
USEUNIT("tut3_3_methodimp.cpp");
//-----
#pragma argsused

int main(int argc, char **argv)
{
    char Choice;
    bool Done;

    // Insert a line to initialise Done correctly.
    Done = false;

    // Initialise the datastructure for employees
    CompanyInitialise();

    // Insert an appropriate condition for the loop.
    while (!Done) // main loop
    {
        // set up menu of choices
        clrscr();
        WriteStringCr("==== Company Staff Database ====");
        WriteStringCr(" ");
        WriteStringCr("Written by : Rifat for Tutorial 3");
        WriteStringCr(" ");
        WriteStringCr("Select Option");
        WriteStringCr(""); // blank line as spacer
        WriteStringCr("A: add new employee");
        WriteStringCr("D: display current employees");
        WriteStringCr("Q: quit");
        // Insert a line to read a character into Choice from the
        keyboard.
        Choice = ReadCharPr("Enter A, D or Q : ");
        // Complete the switch statement so that 'A' and 'D'
        choices do nothing,
        // but 'Q' does stop the loop.
    }
}
```

```

        Choice = Tut3_uppercase(Choice);

        switch (Choice)
        {
            case 'A' : AddNewEmp();
                       break;

            case 'D' : DisplayEmployees();
                       break;

            // final processing to be added to 'q' case later
            case 'Q' : Done = true;
                       break;
        }
    } //end of main loop

    return 0;
}

```

tut3_3_methodimp.cpp

```

//-----
// Methods implementation

#include <vcl\vcl.h>
#include <fstream.h>
#pragma hdrstop
#include "MT262io.h"
#include "tut3_3_methodimp.h"

//-----
void CompanyType::Init(void)
{
    int index;

    // Initialise all Employees data structure

    for (index = 0; index <= MaxEmp; index = index + 1)
    {
        Employee[index].FullName = " ";
        Employee[index].Address = " ";
        Employee[index].Salary = 0;
    }
}

AnsiString CompanyType::AddEmp(AnsiString EmpName, AnsiString
EmpAddress, int EmpSalary)
{
    AnsiString Message;
    Employee[0].FullName = EmpName;
    Employee[0].Address = EmpAddress;
    Employee[0].Salary = EmpSalary;

    Message = "Employee is succesfully added to database.\n";
    return Message;
}

void CompanyType::DispEmp(void)
{

```

```

int index;
index = 0;

while (Employee[index].Salary != 0)
{
    WriteStringCr("");
    WriteString("Employee Number : ");
    WriteIntCr(index+1);
    WriteString("Full Name : ");
    WriteString(Employee[index].FullName);
    WriteString(" Address : ");
    WriteString(Employee[index].Address);
    WriteString(" Salary : ");
    WriteStringCr(Employee[index].Salary);
    index = index + 1;
}

WriteStringCr("Press Enter to continue");
getchar();
}

```

tut3_3_methodimp.h

```

// Class implementation

#define MaxEmp 2

class CompanyType
{
private:

    struct EmployeeType
    {
        int Salary;
        AnsiString Address;
        AnsiString FullName;
    };
    EmployeeType Employee[MaxEmp];

public:

    void Init(void);
    AnsiString AddEmp(AnsiString EmpName, AnsiString EmpAddress, int
EmpSalary);
    void DispEmp(void);
};

```


Tut3_3_funcimp.cpp

```
//-----  
// Function implementation  
  
#include <vc1\vc1.h>  
#pragma hdrstop  
#include "MT262io.h"  
#include "tut3_3_funcimp.h"  
#include "tut3_3_methodimp.h"  
//-----  
CompanyType Company;  
  
char Tut3_uppercase(char selection)  
{  
    if ((selection >= 'a') && (selection <= 'z'))  
        selection = selection - 32;  
  
    return selection;  
}  
  
void CompanyInitialise(void)  
{  
    Company.Init();  
}  
  
void AddNewEmp(void)  
{  
    AnsiString EmpName;  
    AnsiString EmpAddress;  
    int EmpSalary;  
  
    EmpName = " ";  
    EmpAddress = " ";  
    EmpSalary = 0;  
  
    clrscr();  
    EmpName = ReadStringPr("Enter Employee Name: ");  
    EmpAddress = ReadStringPr("Enter Employee Address: ");  
    EmpSalary = ReadIntPr("Enter Employee Salary: ");  
  
    WriteString(Company.AddEmp(EmpName, EmpAddress, EmpSalary));  
}  
  
void DisplayEmployees(void)  
{  
    Company.DispEmp();  
}
```

tut3_3_funcimp.h

```
// Function prototypes  
  
char Tut3_uppercase(char option);  
void CompanyInitialise(void);  
void AddNewEmp(void);  
void DisplayEmployees(void);
```