



Candidate Instructions

Assignment - 202 Version 1.1

Create Software Components Using ‘C++’ - Level 2

Assignment 7262-22-202

Create software components using ‘C++’ Level 2

Candidates are advised to read all instructions carefully before starting work and to check with your assessor, if necessary, to ensure that you have fully understood what is required.

You must, at all times, observe all relevant Health and Safety precautions.

Time allowed 4 hours

Introduction

A scenario is provided below for this assignment, which is made up of 3 tasks:

1. Task A provides a detailed design specification that should be followed by candidates when developing their program.
2. Task B provides presentation criteria that should be followed by candidates when producing their work.
3. Task C lists what the candidate should produce for their assessor.

Scenario

THE CALCULATOR

Your company has been contracted by The Guilds Primary School to develop a small calculator type computer program to assist pupils in promoting their mathematical skills. In principle, the program will take the format of both a calculator and a simple multiplication program.

All entries will be via the keyboard, be validated and the program menu driven. The main menu will give the pupils an option to create a Multiplication Table or to perform a Calculation.

If a Multiplication Table is selected then any positive value table up to the number 20 with up to 15 lines of display may be selected. Both numbers must be validated as per the criteria set out below.

If a Calculation is selected then entry of two positive numbers, the first up to 1000 and the second up to 999. Both numbers must be validated to their range as also set out below. The pupil will be given the option to enter the following operators: +, -, *, / or % (modulus). Depression of any one of these validated keys will cause the result of the calculation to be displayed.

The program must on completion of a Multiplication Table or a Calculation loop back to the main menu awaiting the next command.

As an employee of the company you have been tasked to create a demonstration program which could later be used for this purpose.

Task A

Candidates should use the following detailed specification to fulfil the company's requirements:

1. As part of the specification, the program must be written using functions. Global variables may be declared, but as far as possible, local variables should be used.
2. On executing the program the screen shall clear and a menu displayed with relevant information, from which the pupil or other user can make selection. The menu should be similar to: -

“ Do you want to create a Multiplication Table (M)
or perform a Calculation (C) ?
Note: Only numbers in the ranges shown, or with a
maximum of 6 digits (including any decimal point)
are accepted.

Enter M or C only : “

3. User selection must be validated to be either **C** or **M** (uppercase characters) only.
4. An entry of **M** shall cause the screen to clear and request for a Multiplication Table number such as: -
“ Please enter the Multiplication Table number
Note: The number must be between 1 and 20 only : “
5. This entry must be validated to be in the range of 1 to 20 and validated to be a float number not exceeding 6 digits including any decimal point. If the entry is outside that range then a suitable message should be displayed and the number re-entered. The message should be similar to: -

“ Number not between 1 and 20. Press any key to continue:... “

6. On acceptance of a valid number the screen shall clear and a message requesting the number of lines in the table displayed. This input must also be validated to be in the range of 1 to 15 and either be an integer number or a float number which is converted to an integer number for the purpose of output. As in (5) above a suitable message and out of range message should be displayed.
7. When the second number has been accepted, the screen should automatically clear and the multiplication table neatly displayed with a suitable heading to indicate which table it represents. At the bottom of the screen shall be displayed a message similar to: -

“ Type ‘E’ to exit or any other key to continue “

8. Depression of the letter ‘E’ shall cause the program to terminate after clearing the screen. Any other key shall cause the screen to clear and the main menu at (2) to be displayed awaiting further selection.
9. On selection of **C** the screen to clear and a request for a number displayed such as: -

“ Please enter your first number (0 -> 1000) : “

10. This entry must be validated to be in the range of 0 to 1000 and to be a float number not exceeding 6 digits including any decimal point. An invalid entry shall cause the number entry screen to be repeated.
11. On successful entry, there should be a request for entry of the second number with a message such as: -
“ Please enter your second number (0 -> 999) : “
12. This entry must be validated to be in the range of 0 to 999 and also be a float number not exceeding 6 digits including any decimal point. An invalid entry shall cause the number entry screen to be repeated.
13. After the second number has been accepted, the screen shall clear and display: -
“ Please make your operator selection (+ , - , * , / or %) : “
14. User entry must be validated by a single keystroke to accept only one of these operators.
15. On depression of the operator, the screen shall clear and the full sum displayed with the result.

i.e. *number1 operator number2 = result*
16. If the modulus operator is selected (%) then a message similar to: -
“ This is a modulus division which displays the remainder of an integer division “
17. This should be followed by an output similar to that at (15).
18. At the bottom of the screen should be displayed the same message as at (7) with the same result as (8) above.

Task B

Candidates should follow the criteria below when producing their work:

1. The program conforms to the design specification.
2. The program uses the most appropriate data type(s).
3. Meaningful names are used when declaring variables.
4. The most appropriate method is used to read from the keyboard.
5. The syntax is consistently indented to aid readability.
6. The program syntax uses case sensitivity.
7. The program is commented.

Task C

Candidates should produce the following for their assessor:

1. A printed program listing.
2. A disk containing the program (*the program should be saved under a meaningful Name*).
3. Testing evidence, which compares expected output from the actual run-time output.

Note

- At the conclusion of this assignment, hand all paperwork and disks to the test supervisor.
- Ensure that your name is on the disk (if using a floppy disk) and all documentation.
- If the assignment is taken over more than one period, all floppy disks and paperwork must be returned to the test supervisor at the end of each sitting.