

**BITP 1113: PROGRAMMING TECHNIQUES**  
**1BITS**  
**Lab 5**  
**Repetition**

**Exercise 1**

Write a program to compute and display multiplication table. User will provide the multiplicand and maximum number of multiplier then the program will display the multiplicand times table.

1. Analyse input, process and output.

**Input** : multiplier and multiplicand

**Process** : product = multiplier \* multiplicand;

**Output** : times table

2. Create a new project named Table
3. At **Solution Explorer**, right-click **Source Files** folder and add new item named TimesTable.cpp.
4. Write the following default code:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6      return 0;
7  }
```

5. Insert code to receive the values of multiplier and multiplicand from user (Line 5 to 8).

```
4  int main() {
5      cout << "Set the maximum multiplier: ";
6      cin >> multiplier;
7      cout << "Set the multiplicand";
8      cin >> multiplicand;
9      return 0;
10 }
```

6. Declare variables multiplier and multiplicand to fix the error (Line 5).

```
4  int main() {
5      int multiplier, multiplicand; // Declare variables
6      cout << "Set the maximum multiplier: ";
7      cin >> multiplier;
8      cout << "Set the multiplicand";
9      cin >> multiplicand;
10     return 0;
11 }
```

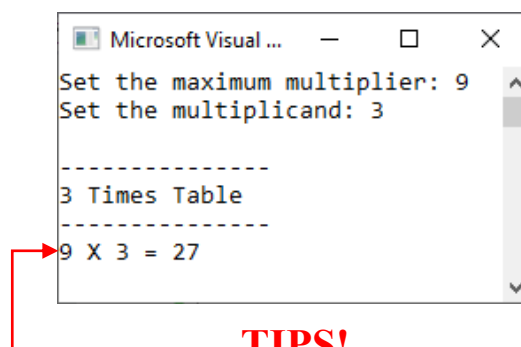
7. Insert code to compute the product and display the times table (Line 12 to 19).

```
5  int main() {
6      int multiplier, multiplicand; // Declare variables
7      cout << "Set the maximum multiplier: ";
8      cin >> multiplier;
9      cout << "Set the multiplicand: ";
10     cin >> multiplicand;
11
12     // Display times table title
13     cout << "\n-----\n";
14     cout << multiplicand << " Times Table\n";
15     cout << "-----\n";
16     // Calculate the product and display the times table
17     product = multiplier * multiplicand;
18     cout << multiplier << " X " << multiplicand
19         << " = " << product << endl;
20
21     return 0;
22 }
```

8. Declare the variable product (Line 5).

```
5  int main() {
6      int multiplier, multiplicand, product; // Declare variables
7      cout << "Set the maximum multiplier: ";
```

9. Compile and run to observe the output.



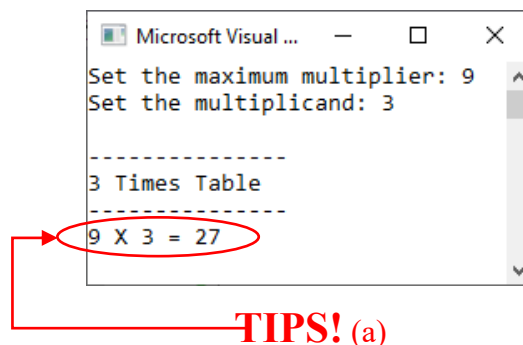
```
Microsoft Visual ...
Set the maximum multiplier: 9
Set the multiplicand: 3

-----
3 Times Table
-----
9 X 3 = 27
```

**TIPS!**

There is a logic error; the output does not show the 3 Times Table instead display the product of two numbers entered by the user. We **MUST** apply **REPETITION** in this code.

10. Identify the repeated output code THEN apply repetition structure by enclosing the repeated output code in { } (Line 16 to 20: Refer **TIPS!(c)** code).



```
Microsoft Visual ...
Set the maximum multiplier: 9
Set the multiplicand: 3

-----
3 Times Table
-----
9 X 3 = 27
```

**TIPS! (a)**

We have to repeat this line of output.

```

12 // Display times table title
13 cout << "\n-----\n";
14 cout << multiplicand << " Times Table\n";
15 cout << "-----\n";
16 // Calculate the product and display the times table
17 product = multiplier * multiplicand;
18 cout << multiplier << " X " << multiplicand
19     << " = " << product << endl;
20
21 return 0;
22 }

```

### TIPS! (b)

These lines of code produce the identified output at TIPS!(a). So, we will apply repetition structure at these lines of code. How to apply repetition structure?

- What is the suitable type of repetition structure? → for loop because we know the number of repetition (1 to maximum number of multiplier).
- What is suitable loop-controller → counter-controlled because we know the number of repeated data.

```

12 // Display times table title
13 cout << "\n-----\n";
14 cout << multiplicand << " Times Table\n";
15 cout << "-----\n";
16 // Calculate the product and display the times table
17 for (int count = 1; count <= multiplier; count++) {
18     product = count * multiplicand;
19     cout << count << " X " << multiplicand
20         << " = " << product << endl;
21 }

```

### TIPS! (c)

1. Apply for loop and declare count as counter-controlled;

- Initialize count=1 because repetition must start from multiplier 1
- Define test expression as count<=multiplier because the multiplication must repeat until the maximum number of multiplier
- Update count by 1 (count++) because the program must produce the product of each multiplier.
- Include {} to enclose the for block of statements

2. Replace the related variables to multiply and display each multiplier.

Original Code

```

product = multiplier * multiplicand;
cout << multiplier << " X " << multiplicand
    << " = " << product << endl;

```

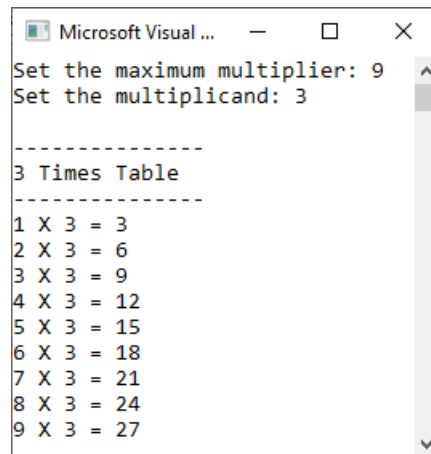
Modified Code

```

product = count * multiplicand;
cout << count << " X " << multiplicand
    << " = " << product << endl;

```

11. Compile and run the program.

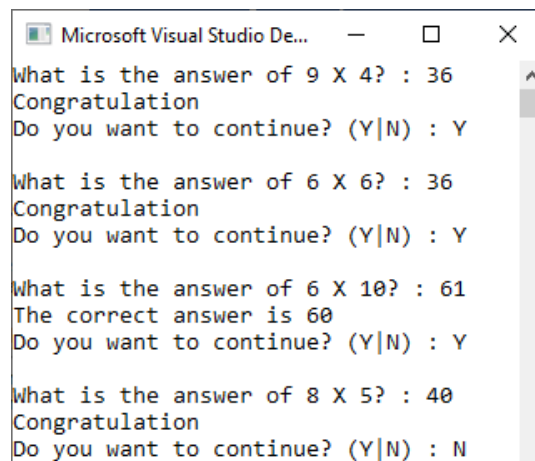


```
Microsoft Visual ...  
Set the maximum multiplier: 9  
Set the multiplicand: 3  
  
-----  
3 Times Table  
-----  
1 X 3 = 3  
2 X 3 = 6  
3 X 3 = 9  
4 X 3 = 12  
5 X 3 = 15  
6 X 3 = 18  
7 X 3 = 21  
8 X 3 = 24  
9 X 3 = 27
```

## Exercise 2

Write a code to create Multiplication Quiz program. The program will generate the question automatically and provide a result after comparing both answer scheme and answer from user. The multiplication involves two random numbers between 1 to 10 (inclusive 1 and 10). Then the program will ask the user either wants to continue or stop answering the quiz.

## Example Output



```
Microsoft Visual Studio De...  
What is the answer of 9 X 4? : 36  
Congratulation  
Do you want to continue? (Y|N) : Y  
  
What is the answer of 6 X 6? : 36  
Congratulation  
Do you want to continue? (Y|N) : Y  
  
What is the answer of 6 X 10? : 61  
The correct answer is 60  
Do you want to continue? (Y|N) : Y  
  
What is the answer of 8 X 5? : 40  
Congratulation  
Do you want to continue? (Y|N) : N
```

1. Analyse input, process and output.

**Input** : userAnswer

**Process** : generate random numbers as operands:

```
number1 = rand() % 10 + 1;  
number2 = rand() % 10 + 1;  
product = number1 * number2;
```

**Output** : result;

```
if correct answer, display "Congratulation".  
if incorrect answer, display the answer scheme.
```

2. Create a new project named Quiz

3. At **Solution Explorer**, right click **Source Files** folder and add new item named Multiplication.cpp.

4. Write the default code.

5. Insert code to generate two random numbers (Line 5 to 9).

```

4  int main() {
5      // Create a random question
6      srand(time(0)); // Initialize a random number generator
7      // Assign random number within range 1 to 10 to the variables
8      number1 = rand() % 10 + 1;
9      number2 = rand() % 10 + 1;

```

**TIPS!**

1. `srand()` sets the seed as a starting point for generating random number. `time` is used to get seconds from system clock, which provide different integer value of seeds since the time is continually changing.

2. Generate random number:

- i) `rand()`: generates random number
- ii) `% 10 + 1`: number between 1 to 10 (inclusive 1 and 10)

6. Declare variables `number1` and `number2` to fix the error (Line 5).

```

4  int main() {
5      int number1, number2; // Declare variables
6      // Create a random question
7      srand(time(0)); // Initialize a random number generator

```

7. Include preprocessor directive for `srand()`, `rand()` and `time()`.

```

1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;

```

8. Insert code to display question and read answer (Line 11 to 14).

```

11 // Display question and read the answer from user
12 cout << "What is the answer of ";
13 cout << number1 << " X " << number2 << "? : ";
14 cin >> userAnswer;
15 return 0;
16 }

```

9. Declare variable `userAnswer` to fix the error (Line 5).

```

4  int main() {
5      int number1, number2, userAnswer; // Declare variables

```

10. Insert code to compare answer scheme and answer from the user then display the result (Line 19 to 24).

```

19 // Compare answers
20 actualAnswer = number1 * number2;
21 if (userAnswer == actualAnswer)
22     cout << "Congratulation";
23 else
24     cout << "The correct answer is " << actualAnswer;
25
26 return 0;
27 }

```

11. Declare variable `actualAnswer` to fix the error (Line 5).

```

6  int main() {
7      int number1, number2, userAnswer, actualAnswer; // Declare variables

```

12. Insert code to ask the user either wants to continue or stop answering (Line 26 to 28).

```
26 // Ask user to continue answering question
27 cout << "Do you want to continue? (Y|N) : ";
28 cin >> choice;
29
30 return 0;
31 }
```

13. Declare variable choice to fix the error (Line 8).

```
6 int main() {
7     int number1, number2, userAnswer, actualAnswer; // Declare variables
8     char choice; // Declare variable
9     // Create a random question
```

14. Identify the repeated output code THEN apply repetition structure by enclosing the repeated output code in { } (Line 9 and Line 31).

```
6 int main() {
7     int number1, number2, userAnswer, actualAnswer; // Declare variables
8     char choice; // Declare variable
9     do { // Start repetition
10         // Create a random question
11         srand(time(0)); // Initialize a random number generator
12         // Assign random number within range 0 to 10 to the variables
13         number1 = rand() % 10 + 1;
14         number2 = rand() % 10 + 1;
15
16         // Display question and read the answer from user
17         cout << "What is the answer of ";
18         cout << number1 << " X " << number2 << "? : ";
19         cin >> userAnswer;
20
21         // Compare answers
22         actualAnswer = number1 * number2;
23         if (userAnswer == actualAnswer)
24             cout << "Congratulation";
25         else
26             cout << "The correct answer is " << actualAnswer;
27
28         // Ask user to continue answering question
29         cout << "Do you want to continue? (Y|N) : ";
30         cin >> choice;
31     } while (choice == 'Y'); // End repetition
32     return 0;
33 }
```

### TIPS!

How to apply repetition structure?

- What is the suitable type of repetition structure? → do...while loop because we do not know the number of repetition and the loop block of statements must be executed at least once.
- What is suitable loop-controller → user-controlled because the user will decide either to continue or stop answering.
- What is the loop test expression? → choice == 'Y' because the loop block of statements will be executed as long as the user choose 'Y'.

15. Compile and run the program. The output will be exactly same as Example Output.

## Lab Attendance Week 5

### Question 1

The Exercise 2 program has been modified as follows but the repetition part is missing. Complete the program to produce an output as shown at Figure 1.

Notes: You may use logical operator to formulate the loop test expression.

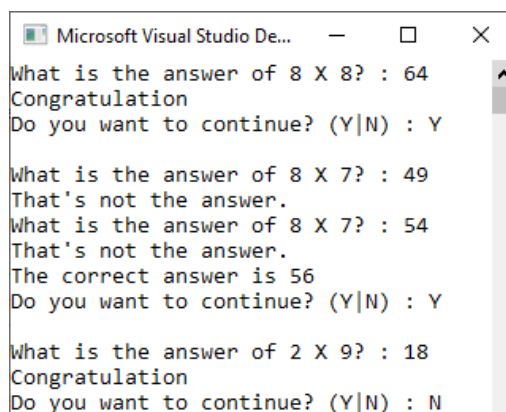
```
int main() {
    int number1, number2, userAnswer, actualAnswer; // Declare variables
    char choice; // Declare variable

    // Create a random question
    srand(time(0)); // Initialize a random number generator
    // Assign random number within range 1 to 10 to the variables
    number1 = rand() % 10 + 1;
    number2 = rand() % 10 + 1;

    // Declare variable to count number of attempts.
    int count = 0;
    // Display question and read the answer from user
    cout << "What is the answer of ";
    cout << number1 << " X " << number2 << "? : ";
    cin >> userAnswer;

    // Compare answers
    actualAnswer = number1 * number2;
    if (userAnswer == actualAnswer)
        cout << "Congratulation";
    else {
        cout << "That's not the answer.\n";
        count++;
    }
    if (count == 2)
        cout << "The correct answer is " << actualAnswer;

    // Ask user to continue answering question
    cout << "\nDo you want to continue? (Y|N) : ";
    cin >> choice;
    cout << endl;
    return 0;
}
```



```
Microsoft Visual Studio De...
What is the answer of 8 X 8? : 64
Congratulation
Do you want to continue? (Y|N) : Y

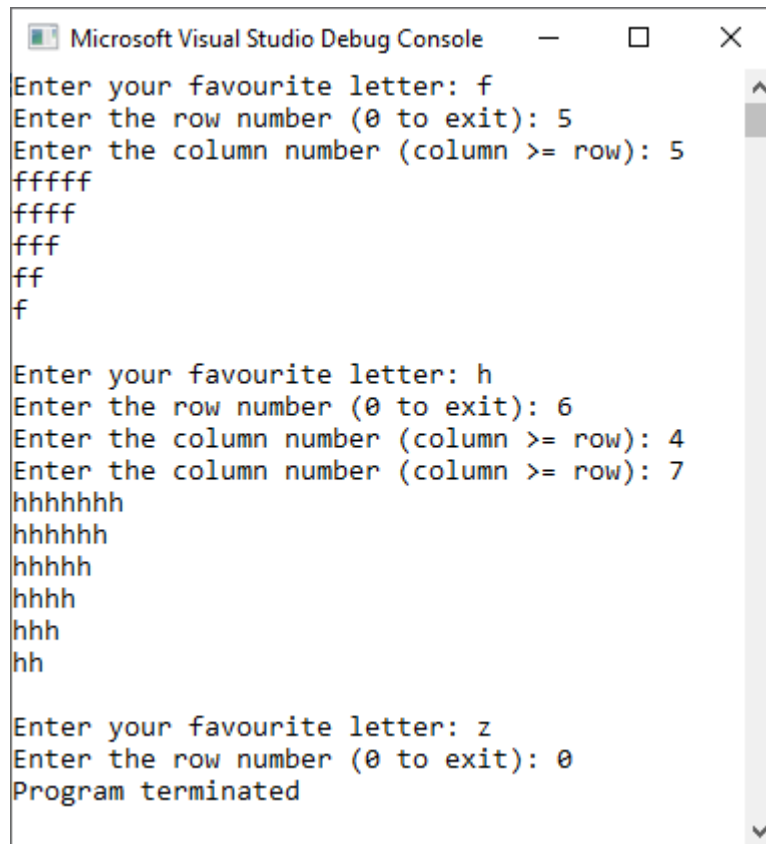
What is the answer of 8 X 7? : 49
That's not the answer.
What is the answer of 8 X 7? : 54
That's not the answer.
The correct answer is 56
Do you want to continue? (Y|N) : Y

What is the answer of 2 X 9? : 18
Congratulation
Do you want to continue? (Y|N) : N
```

Figure 1: Example input/output:

## Question 2

Write a program to produce the following output example.



```
Microsoft Visual Studio Debug Console
Enter your favourite letter: f
Enter the row number (0 to exit): 5
Enter the column number (column >= row): 5
fffff
ffff
fff
ff
f

Enter your favourite letter: h
Enter the row number (0 to exit): 6
Enter the column number (column >= row): 4
Enter the column number (column >= row): 7
hhhhhhh
hhhhhh
hhhhh
hhhh
hhh
hh

Enter your favourite letter: z
Enter the row number (0 to exit): 0
Program terminated
```

Submit this exercise at ULearn before 12.00 p.m. 12 November 2020 (Thursday).

*~Do not lose hope, nor be sad~  
- Al-Quran 3:139*