Topic 4A

Organising Programs with Methods

Topics

Objectives:

- Be able to write programs using methods

Need for Modular Program Design

- In most large software projects, thousands of lines of C# codes are written.
- Codes that are written as one single program are:
  - Difficult, if not impossible, debug if there are any errors
  - Difficult to maintain
- It is therefore, necessary, to break down a complex problem to smaller simpler sub problems of manageable size.
- We then develop algorithms for each of these sub problems. Each algorithm is implemented as a module. In C#, modules are known as **methods**.

Guidelines for Writing methods

- A method must be large enough to perform its task, and must include only the operations that contribute to the performance of *that* task
- The name of the method should describe the work to be done as a single specific task
- As a good naming convention, use a verb followed by 2 to 3 words to describe the task, example:

  CalculateSalesTax

  ComputeAverageMarks

Structure of a Method

return type: void means the method is not returning any value to the calling program.

Name of method

```
void MethodName ( )
{
    statement;
    statement;
}
```

a pair of parentheses or round brackets to contain parameters

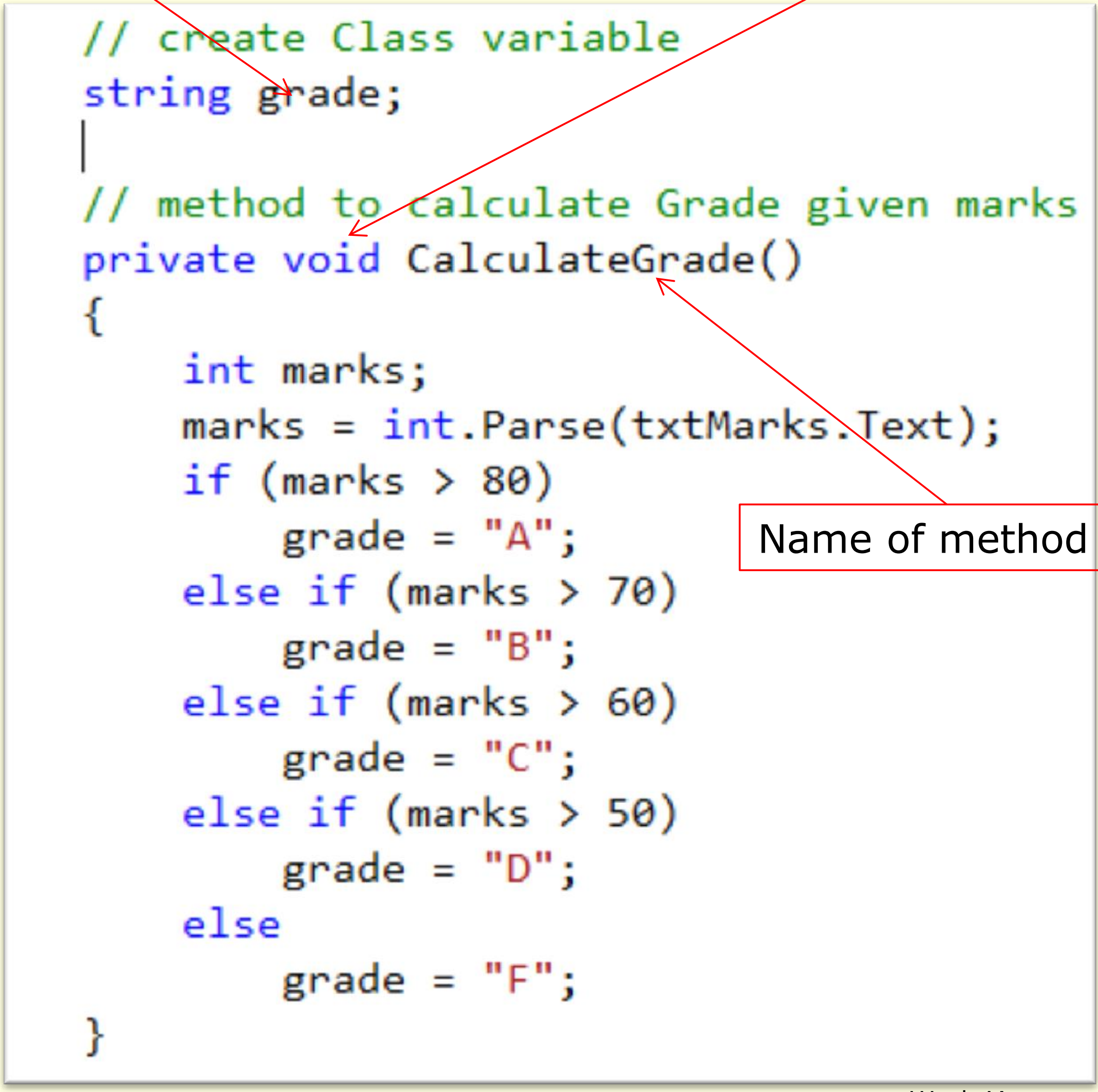
a pair of braces or curly brackets denote the method body containing C# statements

Structure of a Method

- Example:

class variable for storing data which can be access by methods within class

return type: void means the method is not returning any value to the calling program



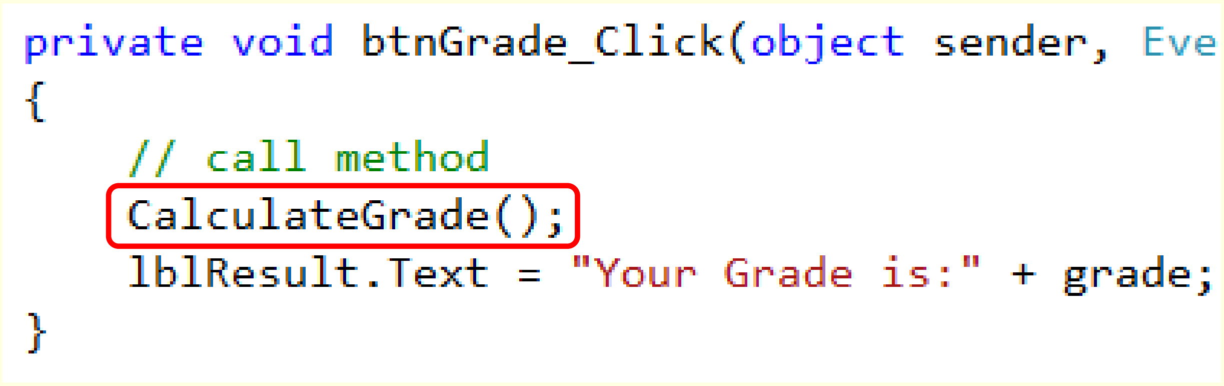
```
// create Class variable
string grade;

// method to calculate Grade given marks
private void CalculateGrade()
{
    int marks;
    marks = int.Parse(txtMarks.Text);
    if (marks > 80)
        grade = "A";
    else if (marks > 70)
        grade = "B";
    else if (marks > 60)
        grade = "C";
    else if (marks > 50)
        grade = "D";
    else
        grade = "F";
}
```

Name of method

How to Call a Method

- ❑ A method is **activated** by calling its name together with 2 brackets ( )
- ❑ Example:



```
private void btnGrade_Click(object sender, Eve
{
    // call method
    CalculateGrade();
    lblResult.Text = "Your Grade is:" + grade;
}
```

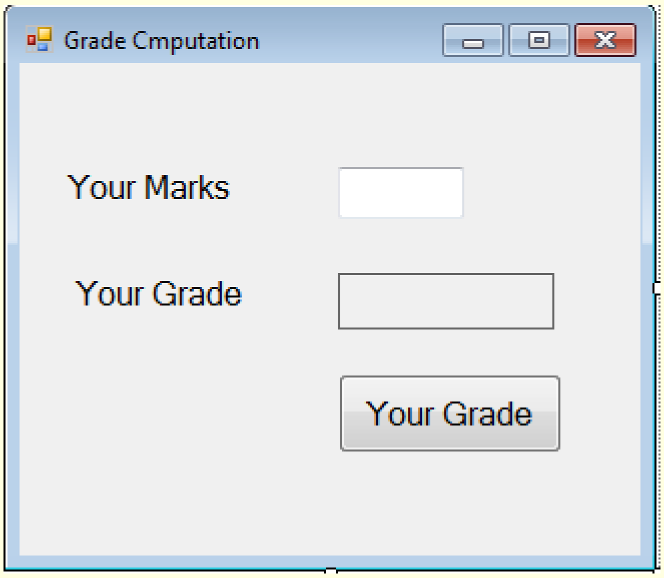
- ❑ Let’s look at an example.

Example 1: Grade Computation

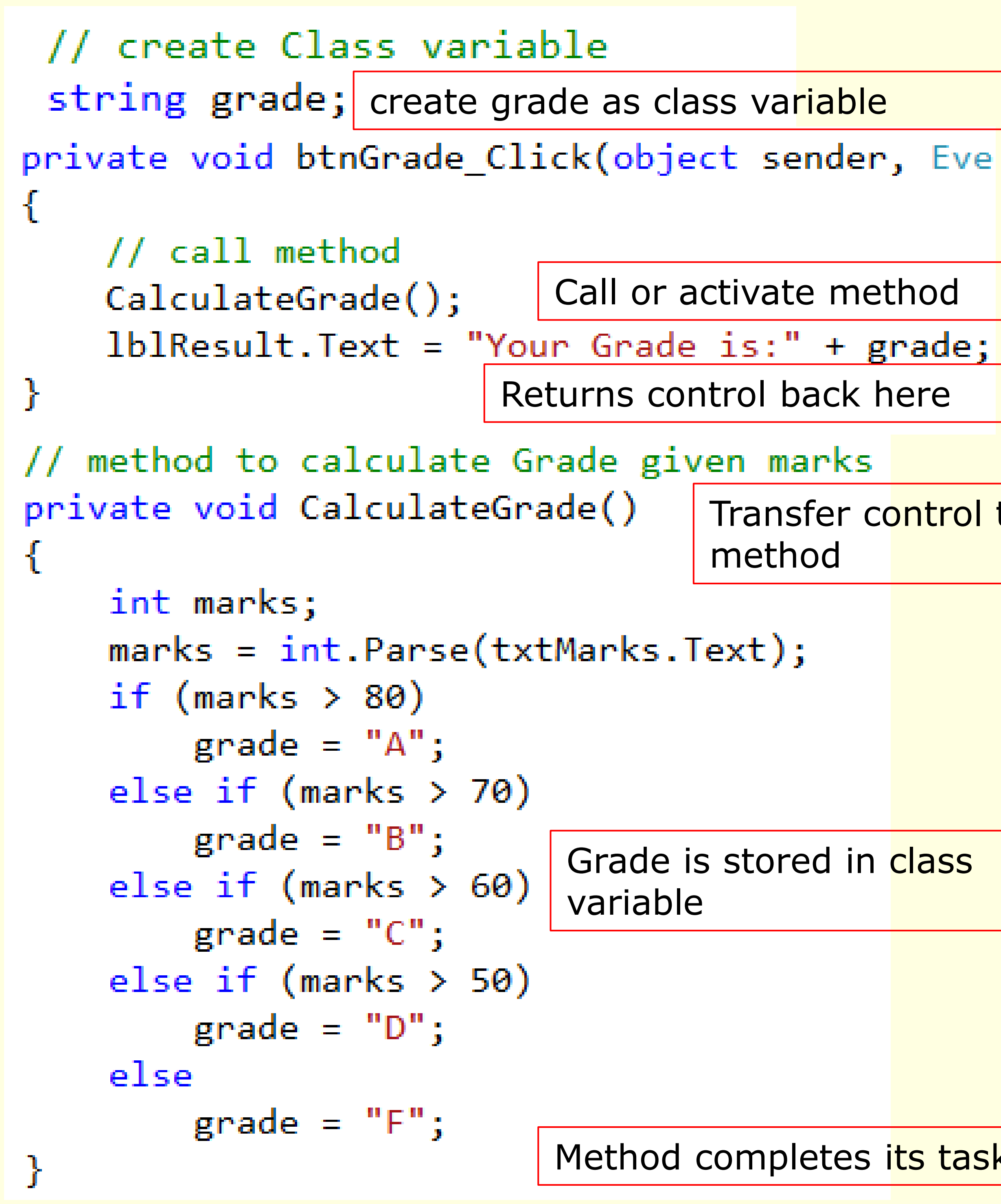
- **Task**: Create a Form to allow the user to enter marks of a student. The Grade will be computed based on following criterion:

Greater than or equal to 80	A
Between 70 to 79 (both inclusive)	B
Between 60 to 69 (both inclusive)	C
Between 50 to 59 (both inclusive)	D
Below 50	F

- **Form Design:**



Example 1: Grade Computation



```
     // create Class variable
     string grade;
    private void btnGrade_Click(object sender, Eve
    {
        // call method
1       CalculateGrade();
5       lblResult.Text = "Your Grade is:" + grade;
    }
    // method to calculate Grade given marks
2   private void CalculateGrade()
    {
        int marks;
        marks = int.Parse(txtMarks.Text);
        if (marks > 80)
            grade = "A";
        else if (marks > 70)
3           grade = "B";
        else if (marks > 60)
            grade = "C";
        else if (marks > 50)
            grade = "D";
        else
            grade = "F";
4   }
```

create grade as class variable

Call or activate method

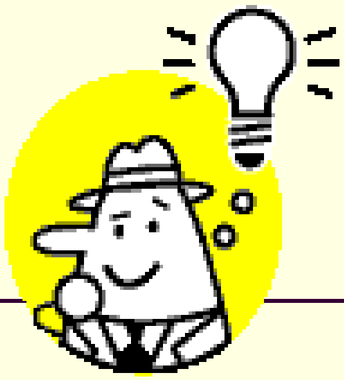
Returns control back here

Transfer control to method

Grade is stored in class variable

Method completes its task

Questions for thought



- Refer to the example1
- Answer the following questions:
  1. Why do we need to create **grade** as a **class variable**?
  2. Will it work if grade were to be a local variable in CalculateGrade() method? Yes / No. Why?
  3. What is the advantage of using a method to Calculate Grade?

More of class variable and method will be covered in Topic 4B

Summary

- Steps in modularization
- Why use modules?
- How to write modules in C# methods
- Features of a method
- Method Definition
  - This contains the C# code to implement the task the module is to accomplish.
- Method Call or Execution
  - For a method to be executed, it must be "called" from another method.

Practical 4A

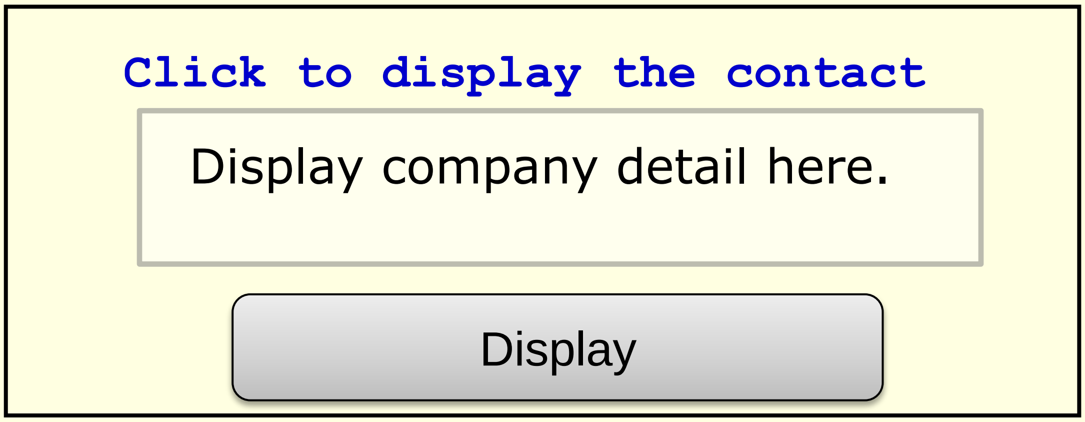
1. A company often needs to display its name, address and telephone contact as follows:

   Zillon Dollar Enterprise Pte Ltd
   11 Holland Road (123456)
   Tel: 6444 8888

   Write a method to display the above 3 lines in a rich text box. Call this method DisplayCompanyDetails.

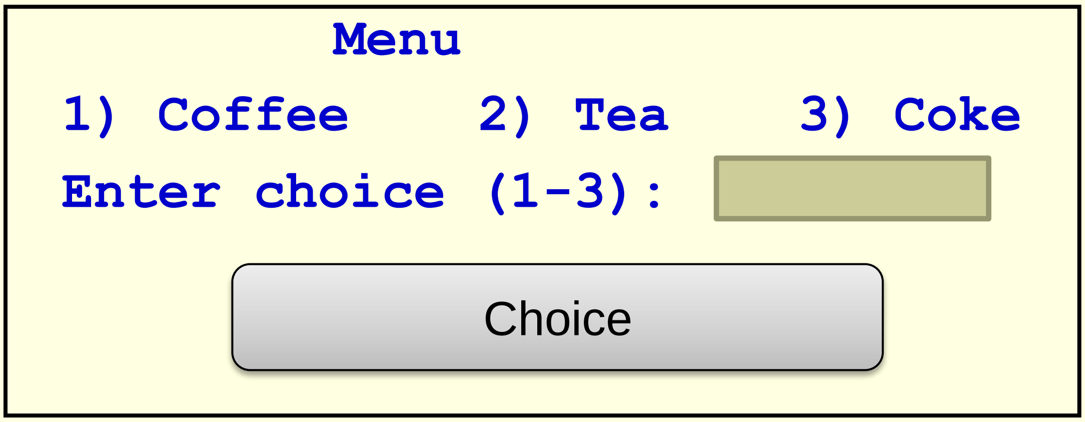
   Create the GUI form to allow user to click on DISPLAY button. It calls the method DisplayCompanyDetails.

Modify the program to display the contact 3 times.
Hint: call the method 3 times.



Practical 4A

2. Write a program to display the menu below and ask user for a choice.



```
Menu
1) Coffee     2) Tea     3) Coke
Enter choice (1-3):
```

When user clicks on **Choice** button, it reads in choice and do one of the following using a Nested-If-Else structure:

If choice is 1, display “Coffee selected” in a message box.

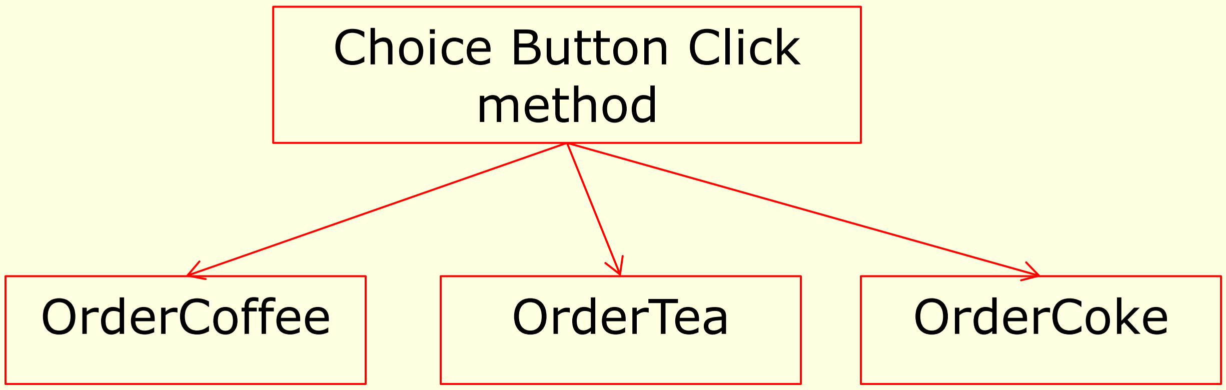
If choice is 2, display “Tea selected” in a message box.

If choice is 3, display “Coke selected” in a message box.

If choice is not 1 or 2 or 3, display “No such choice”.

Practical 4A

3. Refer to Question 2 (In previous slide). Convert the solution to have 4 methods. as shown in the diagram below.



Each box represents a Method.
In each of the OrderXXX method, just display the message “XXX is ordered.”

Example:
In the OrderTea Method, just display the message “Tea is ordered.” in a messagebox

ChoiceButton_ Click Method will call the respective method based on the choice entered.

End of Topic 4A

Organising Programs with Methods