

Topic 3A

Making Comparisons & Choices (Relational Operators, Basic If /else , Check Boxes and Radio Buttons)

Part I

Topics

Objectives:

- ❑ Understand the purpose of relational operators
- ❑ Understand the use of simple if/else Control structure for decision making
- ❑ Understand the use of Check Box and Radio Buttons to allow for user preferences

Introduction

- ❑ Often in our programs, we need to make comparisons between 2 numerical values, 2 strings or 2 char

Example :

Is your age **greater than** 18?

Is your grade **equal to** 'A'?

- ❑ These comparisons uses **relational operators** which we will look at now.

Relational Operators

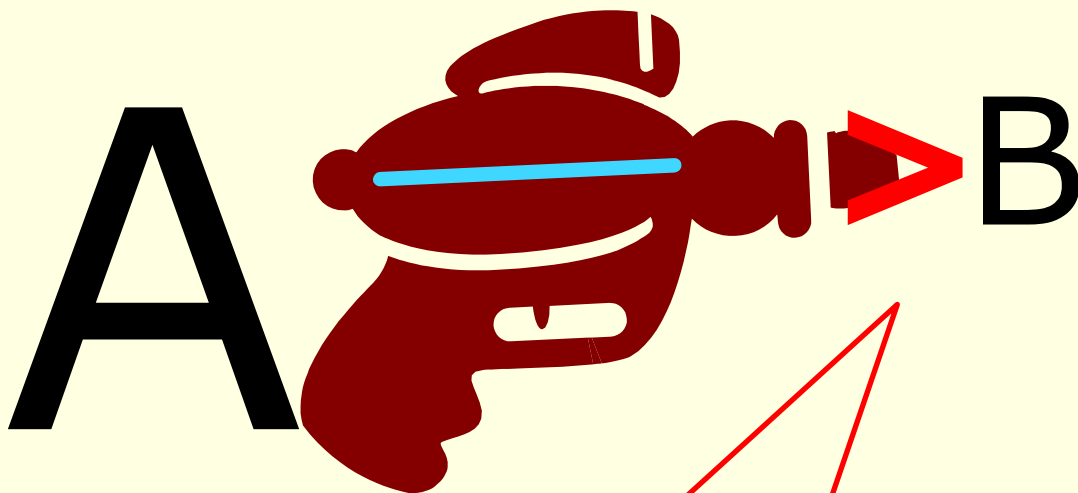
Relational Operators

<	Less than
>	Greater than
==	Equal to
<=	Less than or equal to
>=	Greater than or equal to
!=	Not Equal to

- Relational Operators are used to check whether a condition expression is **TRUE** or **FALSE**
- When you use Relational Operator to make comparison, the result is either **true** or **false**.

TIP: How to remember > sign?

- ❑ $A > B$
 - ❑ Who is bigger?
 - ❑ A or B
 - ❑ **A** is bigger than B, therefore A can fire at poor little B



Arrow symbolizes a **gun**.

Testing you now..

☐ $16 > 15$

True

☐ True or False

☐ $14 < 15$

True

☐ True or False

☐ $15 > 15$

False

☐ True or False

☐ $14 < 14$

False

☐ True or False

☐ $15 \geq 15$

True

☐ True or False

☐ $14 \leq 14$

True

☐ True or False

Relational Operators

- ❑ Let's look at some examples for **numerical values** :

```
int age1 = 18;
int age2 = 20
bool result; // bool data type stores True or False
result = age1 > age2; // age1 greater than age2
lblLabel.Text =result.toString(); // Displays False
```

```
float height1 = 1.75f;
float height2 = 1.5f;
bool result = height1 < height2; // height2 less
than height1
lblLabel.Text =result.toString(); // Displays False
```

```
int score1 = 10;
int score2 = 20
bool result;
result = score1 != score2; // score1 not equal to
score2
lblLabel.Text =result.toString(); // Displays True
```

Relational Operators

- ❑ Let's look at some examples for **strings** and **char** :

```
char let1 = 'A';  
char let2 = 'a';  
bool result;  
result= let1 != let2; // let1 not equal to let2  
lblLabel.Text =result.toString(); // Displays True.  
// Why?
```

```
string name1 = "Pete";  
string name2 = "pete"  
bool result;  
result = name1 == name2; // name1 equal to  
name2  
lblLabel.Text =result.toString(); // Displays False  
// Why?
```


Order of Precedence

Order of precedence

()	• Left to Right	1
+ - (unary)	• Left to Right	2
* / %	• Left to Right	3
+ - (binary)	• Left to Right	4
< > <= >=	• Left to Right	5
== !=	• Left to Right	6
=	• Right to Left	7

Example

$A > B + 2$ is the same as $A > (B+2)$

Making decisions using if/else

- ❑ Recall that a computer can perform various arithmetic operations.
- ❑ Another task that a computer does well is to **evaluate a condition and select different alternatives**
- ❑ The C# syntax for if statement is:

if (condition)

do statement1

} Statement1 is executed
if condition is TRUE

optional { **else**

do statement2

} Statement2 is
executed if
condition is False

- ❑ Let's look at examples:

Making decisions using if/else

```
int num1 = 15;
int num2 = 10;

if (num1 > num2)
    lblResult.Text = "True"; // Displays True
else
    lblResult.Text = "False";
```

```
int num1 = 10;
int num2 = 15;

if (num1 == num2)
    lblResult.Text = "True";
else
    lblResult.Text = "False"; // Displays False
```



- == is a relational operator
- Here, we are comparing whether num2 is equal to num1.
- Do NOT be confused with the assignment operator =

```
if (num1 = num2) // WRONG!
    lblResult.Text = "True";
```

if Control Structure

- ❑ In the previous examples, there is only **one statement** after the if or else clause.

```
int num1 = 15;  
int num2 = 10;  
  
if (num1 > num2)  
    lblResult.Text = "True"; // Displays True  
else  
    lblResult.Text = "False";
```

- ❑ Let's look at a more general form of the if/else control structure.

Making decisions using if/else

- The general form of if-else implemented in C# is:

```
if (condition is true)
{
    statement1;
    statement2;
} else
{
    statement3;
    statement4;
}
```

}

else
{
Is optional
}

Example :

```
int marks= 60;
char grade;
```

```
if (marks >= 50)
{
    grade = 'P';
    lblDisplay.Text ="Good! You passed."
}
else
    grade = 'F';
```

Making decisions using if/else

- We may also have another if/else within another if/else block. Example:

```
if (condition1 is true)
{
    statement1;
    statement2;
}else
{
    if( condition2 is true)
    {
        statement3;
        statement4;
    }else
    {
        statement5;
        statement6;
    }
}
```

Making decisions using if/else

□ Example:

```
int secretNumber = 8;

int guessNumber = int.Parse(txtGuessNumber.Text);

// guess is correct
if (guessNumber == secretNumber)
{
    lblDisplay.Text = "Well Done";
}
else // guess is too high or low
{
    // guess is too high
    if (guessNumber > secretNumber)
        lblDisplay.Text = "Too High";
    else
        lblDisplay.Text = "Too Low";
}
```

Here we use IF/ELSE statement within another IF/ELSE statement block. It can be implemented using **else if** statement too

Making decisions using **else if statement version**

□ Example:

```
int secretNumber = 8;

int guessNumber = int.Parse(txtGuessNumber.Text);

// guess is correct
if (guessNumber == secretNumber)
{
    lblDisplay.Text = "Well Done";
}
else if (guessNumber > secretNumber)
// guess is too high
{
    lblDisplay.Text = "Too High";
}
else
{
    lblDisplay.Text = "Too Low";
}
```

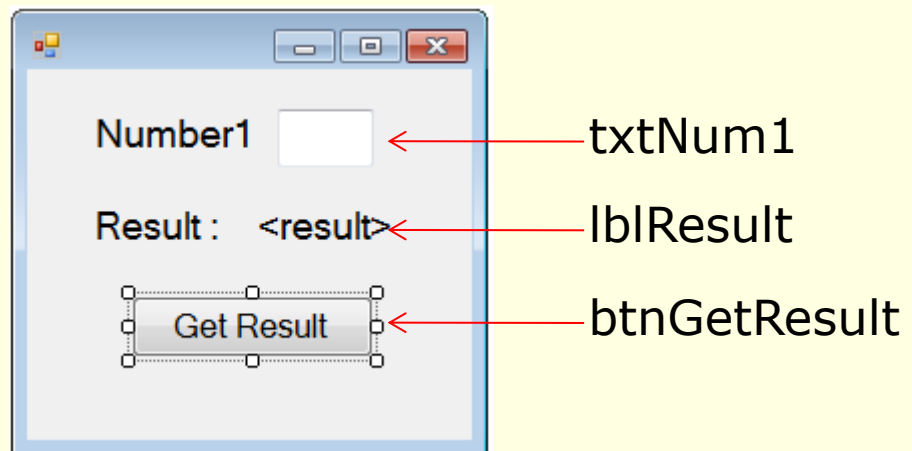
Here we use **else if** statement. .
MORE will be covered in Week 3B.

Example 1: Program to check if a number is divisible by 3

- ❑ **Task:** Create a Form to accept a numeric value from the user.

When the user enters a number and clicks the button, the program will display whether the number is divisible by 3.

- ❑ **Form Design:**



Pseudo code (*elaborate following*):

Read input.

Do calculation.

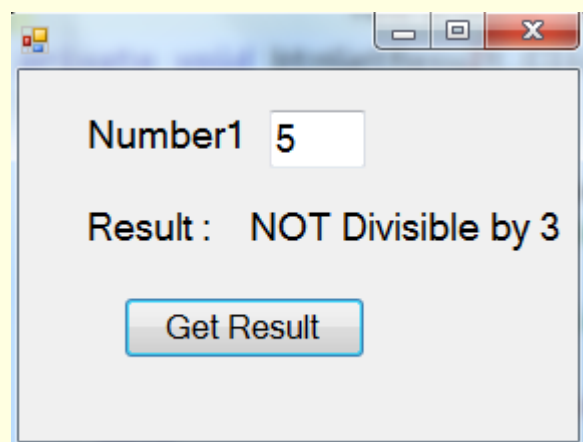
Display output.

Example 1: Program to check if a number is divisible by 3

- ❑ Code: Note : this is an **equal to** comparator. You should **NOT** use `if (num%3 = 0)`

```
private void btnGetResult_Click(object sender, EventArgs e)
{
    int num = int.Parse(txtNum1.Text);
    // if there is no remainder,
    // it is divisible by 3
    // remember % means modulus
    if (num %3 == 0)
        lblResult.Text = "Divisible by 3";
    else
        lblResult.Text = "NOT Divisible by 3";
}
```

- ❑ Sample Output:



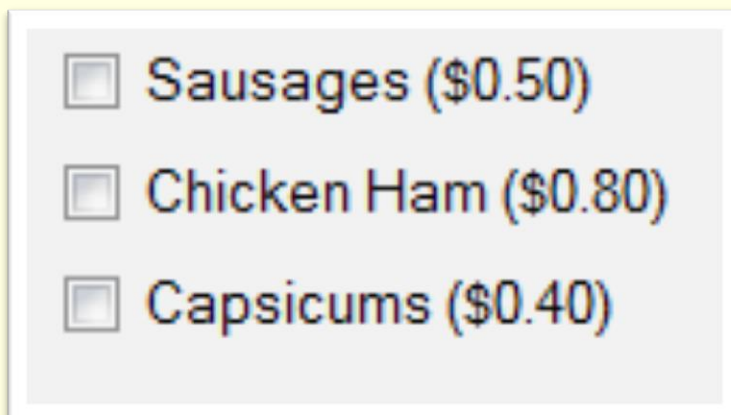
Check Box and Radio Buttons

Controls for accepting user choices

- ❑ In this section, we will introduce 2 controls: Check Box and Radio Button

❑ Check Box

- ❑ A Check Box which consists of a small square presents with a user a yes/no choice.
- ❑ The Checked property of the Check Box is False if the square is empty and True when it is checked.
- ❑ In the example below, 3 Check Boxes are used to allow the user to select the choices he wishes:



☐ Sausages (\$0.50)

☐ Chicken Ham (\$0.80)

☐ Capsicums (\$0.40)

Check Box and Radio Buttons

Controls for accepting user choices

☐ Check Box

- ☐ It accepts One or more selections
- ☐ Let's see how this works using this example:

☐ Sausages (\$0.50)

☒ Chicken Ham (\$0.80)

☒ Capsicums (\$0.40)

Chicken Ham & Capsicum are selected

Total Price **\$1.20**

Calculate

Pseudo code:

If Sausages is selected, add \$0.50 to total price.
If Chicken ham is selected, add \$0.80 to total price.
etc.. etc..

Check Box and Radio Buttons

Controls for accepting user choices

```
private void btnCalculate_Click(object sender, EventArgs e)
{
    float totalPrice = 0.0f;

    1 if (chkSausages.Checked == true)
        totalPrice = totalPrice + 0.50f;

    2 if (chkChickenHam.Checked == true)
        totalPrice = totalPrice + 0.80f;

    3 if (chkCapsicums.Checked == true)
        totalPrice = totalPrice + 0.40f;

    lblTotalPrice.Text = totalPrice.ToString("C");
}
```

- ❑ 3 sets IF statements are used to check the CHECK BOX control one by one. At any time, all 3 or 2 or 1 can be selected.

- 1
 - ❑ IF user selects chkSaugage, compute the totalPrice with sausages added (0.50)
- 2
 - ❑ IF user selects chkChickenHam, compute the totalPrice with chicken ham added (0.80)
- 3
 - ❑ IF user selects chkCapsicums, compute the totalPrice with capsicums added (0.40)

Example 2: Pizza Ordering System

- ❑ **Objective:** Create a Form to accept following:
 - ❑ Quantity of pizza
 - ❑ Choices of none or more of toppings (Sausages, Chicken Ham, Capsicum) . **This is a ONE TIME charge (ie. same for any quantity ordered)**
- ❑ When the button is clicked, the total price is calculated.

- ❑ **Form Design:**

The screenshot shows a Windows application window titled "Kampong Pizza". Inside the window, there is a picture of a chef holding a pizza. Below the picture, the text "Price per 20 inch Pizza" is followed by "\$15.50". To the right of this, there is a text box labeled "How many?". Below the text box, there is a text box labeled "Total Price". To the right of the "Total Price" text box, there is a button labeled "Calculate". To the right of the "Calculate" button, there is a group box titled "Toppings" containing three checkboxes: "Sausages (\$0.50)", "Chicken Ham (\$0.80)", and "Capsicums (\$0.40)".

Annotations with arrows point to the following controls:

- txtQty** points to the "How many?" text box.
- lblTotalPrice** points to the "Total Price" text box.
- btnCalculate** points to the "Calculate" button.
- chkSausage** points to the "Sausages (\$0.50)" checkbox.
- chkHam** points to the "Chicken Ham (\$0.80)" checkbox.
- chkCapsicum** points to the "Capsicums (\$0.40)" checkbox.

Example 2: Pizza Ordering System - Pseudo code



The screenshot shows a window titled "Kampong Pizza". Inside the window, there is a photo of a chef holding a pizza. Below the photo, the text "Price per 20 inch Pizza" is followed by "\$15.50". To the left of a text input field is the label "How many?". To the right of the input field is a "Calculate" button. Below the input field is a "Total Price" label followed by another text input field. To the right of these fields is a "Toppings" section with three checkboxes: "Sausages (\$0.50)", "Chicken Ham (\$0.80)", and "Capsicums (\$0.40)".

//Read inputs

Read quantity.

Read check box sausage.

Read check box chicken ham.

Read check box capsicums.

//Do calculations

total price = quantity x 15.50

If sausage selected, add 0.50 to total price

If chicken ham selected, add 0.80 to total price

If capsicum selected, add 0.40 to total price

//Display results

Display total price

Example 2: Pizza Ordering System

Code:

```
private void btnCalculate_Click(object sender, EventArgs e)
{
    float totalPrice = 0.0f;
    float basicPrice = 15.50f;
    int quantity;

    // Get qty ordered
    quantity = int.Parse(txtQuantity.Text);
    totalPrice = basicPrice * quantity;

    if (chkSausages.Checked == true)
        totalPrice = totalPrice + 0.50f;

    if (chkChickenHam.Checked == true)
        totalPrice = totalPrice + 0.80f;


    if (chkCapsicums.Checked == true)
        totalPrice = totalPrice + 0.40f;

    lblTotalPrice.Text = totalPrice.ToString("C");
}
```


Example 2: Pizza Ordering System

□ Sample Output:

Kampong Pizza



Price per 20 inch Pizza **\$15.50**

How many?

Total Price **\$32.30**

Toppings

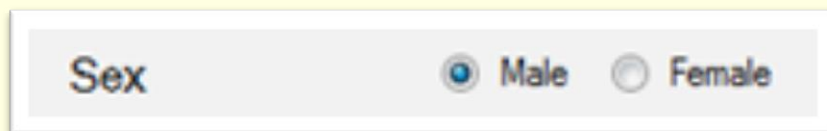
- ☒ Sausages (\$0.50)
- ☒ Chicken Ham (\$0.80)
- ☐ Capsicums (\$0.40)

Check Box and Radio Buttons

Controls for accepting user choices

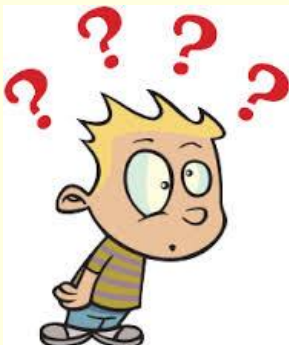
☐ Radio Button

- ☐ Radio Buttons are used to select a **single** choice from several options.
- ☐ In the previous example, if radio buttons were used the user can select either **None** or **one** of any of the 3 choices only (Sausages, Chicken Ham or Capsicum)
- ☐ Let's look at an example next.



Sex ☒ Male ☐ Female

What is the difference between Radio Button and Check Box?



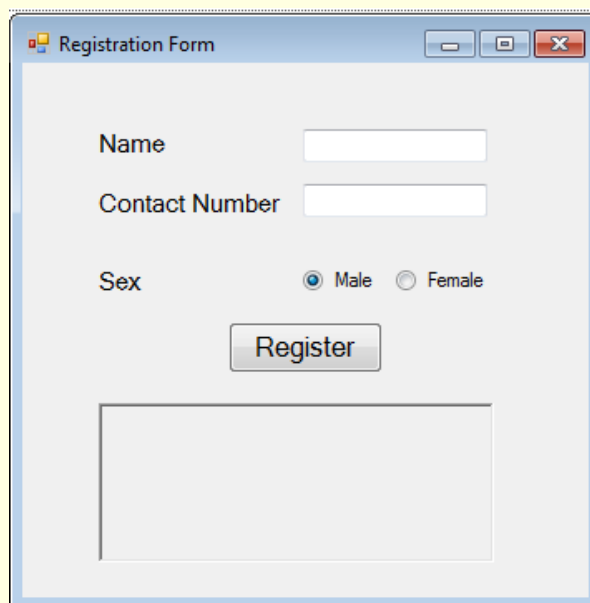
Example 3: Using Radio Buttons

❑ **Objective:** Create a Registration Form to accept the following:

- ❑ Name
- ❑ Contact Number
- ❑ Selection of Sex (Male or Female)

When the user enters the details, and click on the button, the details will be appended to existing records and displayed

❑ **Form Design:**



The image shows a graphical user interface for a registration form. It includes text labels for 'Name', 'Contact Number', and 'Sex'. The 'Name' and 'Contact Number' labels are followed by empty text input boxes. The 'Sex' label is followed by two radio buttons, one labeled 'Male' (which is selected) and one labeled 'Female'. Below these inputs is a button labeled 'Register'. At the bottom of the form area is a large, empty rectangular box, presumably for displaying a list of registered users.

Example 3: Using Radio Buttons – Pseudo code

Registration Form

Name

Contact Number

Sex ☒ Male ☐ Female

Read name.

Read contact number.

{Read radio button sex}

If Male selected, set gender to male

else set gender to female

Form display string with name, contact number and gender.

Display result.

Check Box and Radio Buttons

Controls for accepting user choices

Code

```
private void btnRegister_Click(object sender, EventArgs e)
{
    string sex;
    if (rdoMale.Checked == true)
    {
        sex = "Male";
    }
    else
    {
        sex = "Female";
    }

    rtbDisplay.AppendText("Registration Details:" + Environment.NewLine);
    rtbDisplay.AppendText("=====" + Environment.NewLine);
    rtbDisplay.AppendText("Name:".PadRight(10) + txtName.Text +
        Environment.NewLine);
    rtbDisplay.AppendText("Contact:".PadRight(10) +
        txtContact.Text + Environment.NewLine);
    rtbDisplay.AppendText("Sex:".PadRight(10) + sex);
}
```

1

TO DO : Explain what code does

2

Check Box and Radio Buttons

Controls for accepting user choices

Sample Output:

Registration Form

Name

Contact Number

Sex ☒ Male ☐ Female

Registration Details:
=====

Name: Peter Tan
Contact: 67771234
Sex: Male

Summary

❑ Relational Operators

<	Less than
>	Greater than
==	Equal to
<=	Less than or equal to
>=	Greater than or equal to
!=	Not Equal to

- ❑ Use to compare numerical values, strings or char
- ❑ Comparison result is either True or False.

Summary

❑ If /Else Structure

- ❑ The if/else control enables the computer to do different actions depending on certain Test Conditions.
- ❑ The key control structure is:

```
if ( test is true )  
{  
    perform these statements;  
}  
else // if test is false  
{  
    perform these statements  
    if test is false;  
}
```


Summary

- ❑ You should enclose all C# statements after a if / else control structure with **{ }**

Example:

```
if ( age > 14 )  
{  
    bool discount = false;  
    txtResult.text = "No discount given";  
}  
else  
{  
    bool discount = true;  
    txtResult.text = "Discount valid";  
}
```

- ❑ However, the { } may be omitted if there is ONLY ONE statement after if / else control structure.

Example:

```
if ( age <= 18 )  
    txtResult.text = "You are under age";
```

- ❑ We strongly recommend students to **INCLUDE** the { } braces at all times.

Summary

- ☐ Check Boxes are used to select **none, 1 or more** choices from several choices.
- ☐ Radio Buttons are used to select a **single** choice from several options.

Open up the starter solution given to you. The form , controls and Partial codes are provided. Complete the solution.

```

1a) If ( Password is equal to "teddybear")
    Display "Correct"
Else
    Display "Wrong"

```

```
private void btnTest_Click(object sender, EventArgs e)
{
    string password = "teddybear";

    if (password == "teddybear")
    {
        lblDisplay.Text = "Correct";
    }
    else
    {
        lblDisplay.Text = "Wrong";
    }
}
```

Practical 3A Guided Exercise

1b)

If (Student age is greater than or equal to 18)
Display "Enter"

Else

Display "You are underage"

1c)

If(age is less than 55)
Display "You are eligible"

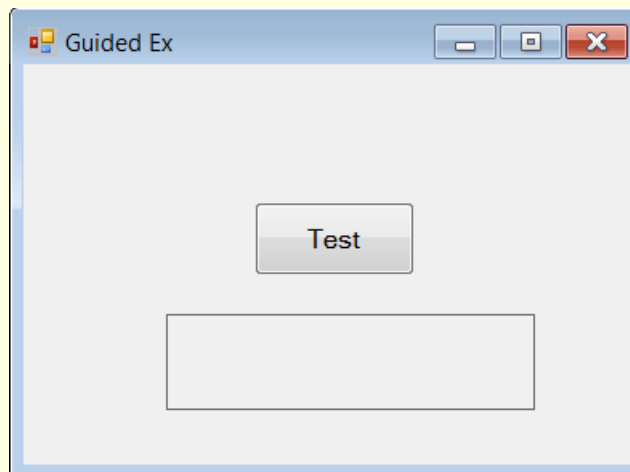
Else

Display "Please report with your checkup first"

Run the program by clicking on the test button. Observe the result displayed

Change the variable's value, run and observe the result displayed

GUI Form Design :

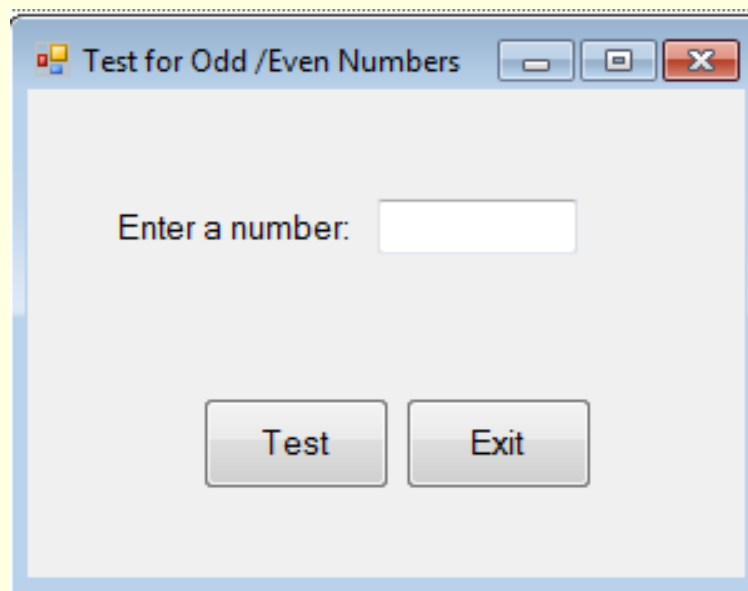


Practical 3A

2. Determine whether a number is odd or even

- ❑ **Task:** Create a Form to accept a number. When the button is clicked, display message “Number is Odd” or “Number is Even”. Starter files are given to you. [Hint: A number is even if the result of $(num1 \% 2)$ is zero]. Use frmOddEven given to you.

❑ GUI Form Design :



Practical 3A

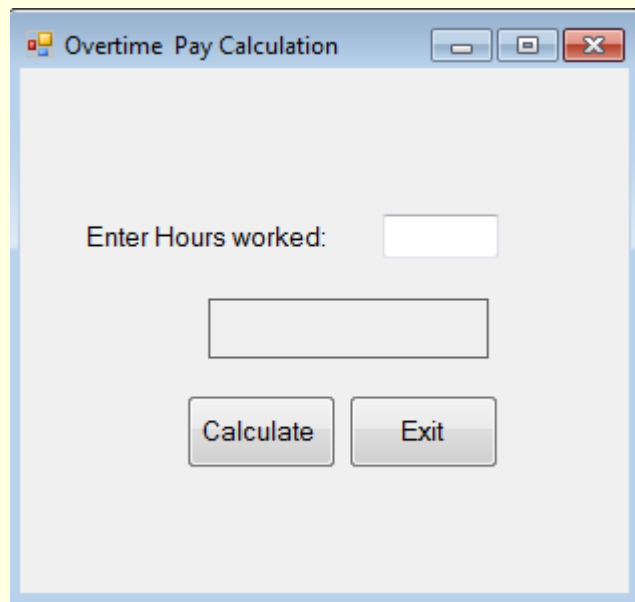
3. Calculate pay of employee

- ❑ **Task:** Create a Form to accept the hours worked by an employee. When the button is clicked, his/her pay will be calculated based on the following:

<u>Hours worked</u>	<u>Hourly Rate</u>
First 44 hours	: \$15.50 /hour
Overtime beyond 44 hours	: \$25.50/hour

Use frmOvertime given to you.

- ❑ **GUI Form Design :**



The screenshot shows a Windows-style application window titled "Overtime Pay Calculation". Inside the window, there is a text label "Enter Hours worked:" followed by a text input field. Below this, there is another empty text input field. At the bottom of the form, there are two buttons: "Calculate" and "Exit".

Practical 3A

4. Guessing Game

- ❑ **Task:** Create a Form to accept a number. When the button is clicked, check whether the number is equal to a secret number between 1 to 10. You assign a generation of the secret number is already given to you in the Form frmGuess. One of three messages will appear:
 - ❑ You guess correctly
 - ❑ Your guess is too high
 - ❑ Your guess is too low.

Hint :

- ❑ Create a **class variable** `int secretNumber`.
- ❑ In the Form Load method, add in the following code to assign a random number between 1 to 10 to `secretNumber`:

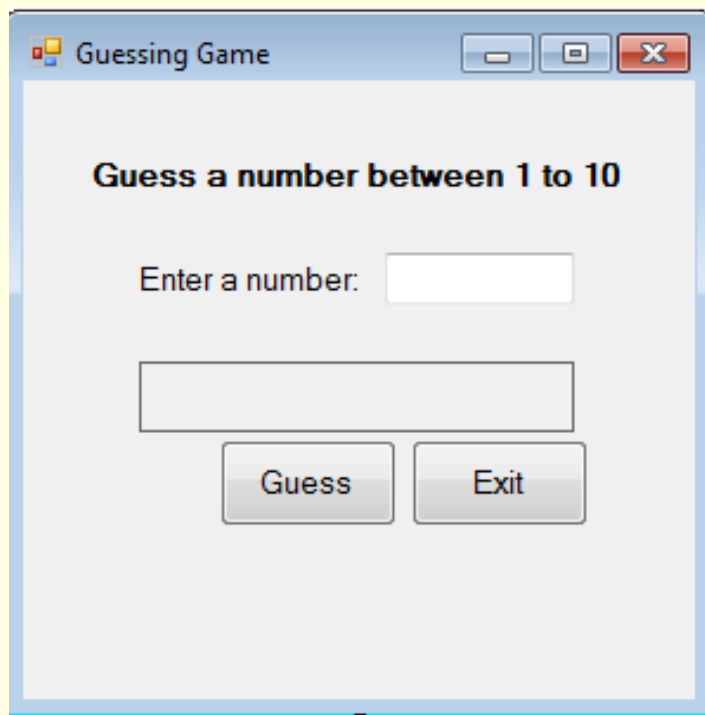
```
Random rnd = new Random();  
secretNumber = rnd.Next(1, 10);
```

Form Load method will be called when user first launch the application. It is used to initialize some settings.

Practical 3A

4. Guessing Game

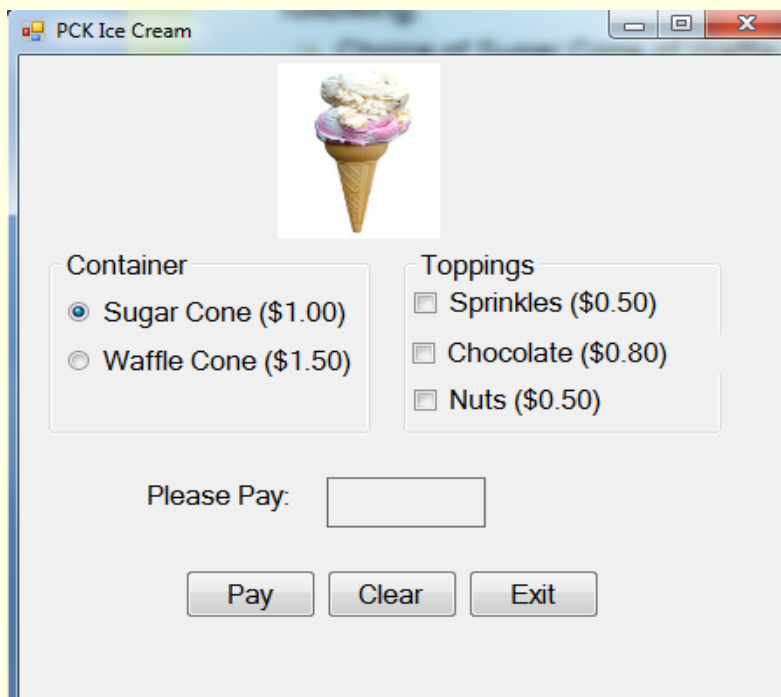
- ❑ To display the right message, you need to use **another if/else within the if/else block** or **else if statement** for the conditional checking (refer to page 14,15,16)
- ❑ **GUI Form Design :**



Practical 3A

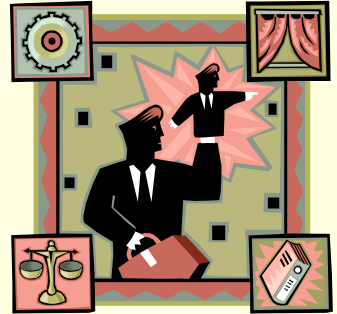
5. PCK Ice Cream Ordering Form

- ☐ **Task:** Create a Form to accept following:
 - ☐ Choice of Sugar Cone or Waffle Cone
 - ☐ Choices of none or more of toppings (sprinkles, chocolates or nuts)
- ☐ When the button is clicked, the total price is calculated. Use the prices stated in the form. The basic price of the ice cream without add on is \$2.50.
- ☐ **GUI Form Design:**



The screenshot shows a Windows-style application window titled "PCK Ice Cream". At the top center is an image of a double-scoop ice cream cone. Below the image are two groups of options. The "Container" group has two radio buttons: "Sugar Cone (\$1.00)" (selected) and "Waffle Cone (\$1.50)". The "Toppings" group has three checkboxes: "Sprinkles (\$0.50)", "Chocolate (\$0.80)", and "Nuts (\$0.50)". Below these groups is a text label "Please Pay:" followed by an empty rectangular input field. At the bottom are three buttons: "Pay", "Clear", and "Exit".

End of Topic 3A



Making Comparisons & Choices (Relational Operators, Basic If /else , Check Boxes and Radio Buttons)

Part I