INF 154
PRACTICAL 5
2023

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

At the end this practical you should be able…

1. Create programs with if / switch statements.
2. Use the random number generator.
3. Use the OpenFileDialog control.
4. Use and understand the methods provided to you by the .NET framework.

What are we going to do?

- Prac 5a: Completed in class.
- Prac 5b: Completed in class.
- Prac 5c: Take home practical.

Practical Exercise 5a

Objective: We would like to generate a random number, then have that number decide the weekday displayed. For example the number 2 is generated then the corresponding day, “Tuesday” would be displayed.

Practical Exercise 5a

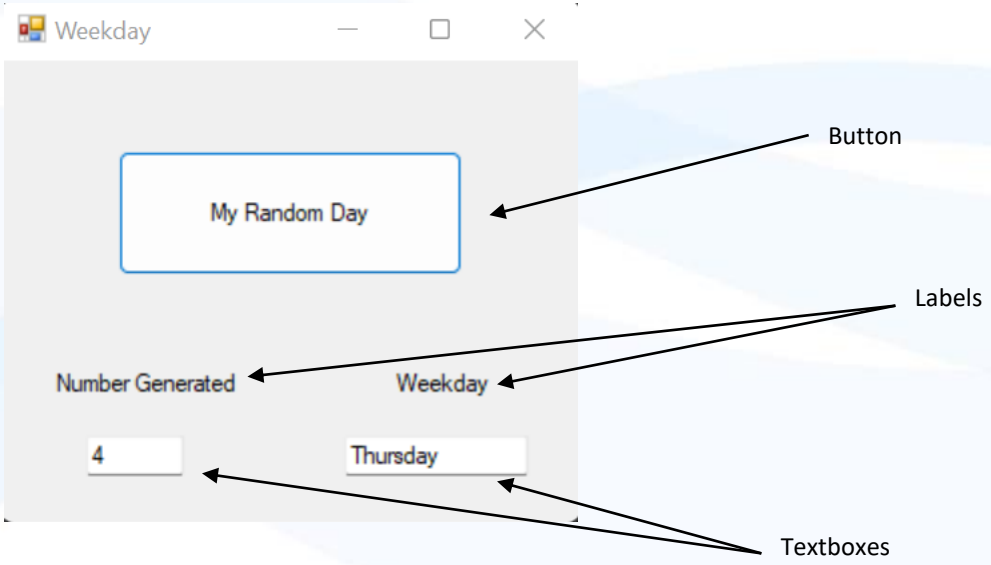
Remember to ask yourself the following questions:

- Inputs and Obtaining Data:
  - Is the user providing us with information, if so, how?
  - Once I have the interface to obtain the data, do I need to manipulate it?
- Processing:
  - How do I solve the problem?
    - » What formulas do I need?
    - » What steps are required?
- Outputs:
  - How are we displaying the information?
  - Do we have to manipulate the data again?

Practical Exercise 5a

The interface:

Now that we have an understanding of the program we would like to implement. We can design the following interface.



Practical Exercise 5a

The code:

- Declaring variables & Random number generation:

```
22          // Declare variables
23          int weekdayNum;
24          string weekday;
25          // Create a instance of the Random Class
26          Random randomNum = new Random();
27          // Generate a random integer between 1 and 7 inclusive
28          weekdayNum = randomNum.Next(1,8);
29          txtRndNum.Text = weekdayNum.ToString();
```

- Processes & Output:

```
31          // Create the switch statement
32          switch (weekdayNum)
33          {
34              case 1:
35                  weekday = "Monday";
36                  break;
37              case 2:
38                  weekday = "Tuesday";
39                  break;
40              case 3:
41                  weekday = "Wednesday";
42                  break;
43              case 4:
44                  weekday = "Thursday";
45                  break;
46              case 5:
47                  weekday = "Friday";
48                  break;
49              case 6:
50                  weekday = "Saturday";
51                  break;
52              case 7:
53                  weekday = "Sunday";
54                  break;
55              default:
56                  weekday = "Invalid weekday";
57                  break;
58          }
59
60          txtWeekday.Text = weekday;
```

Practical Exercise 5b

Objective: We would like to create a password generator. The user will enter a word and select a number between 1 and 26. The system will then combine the inputs, creating a password.

Practical Exercise 5b

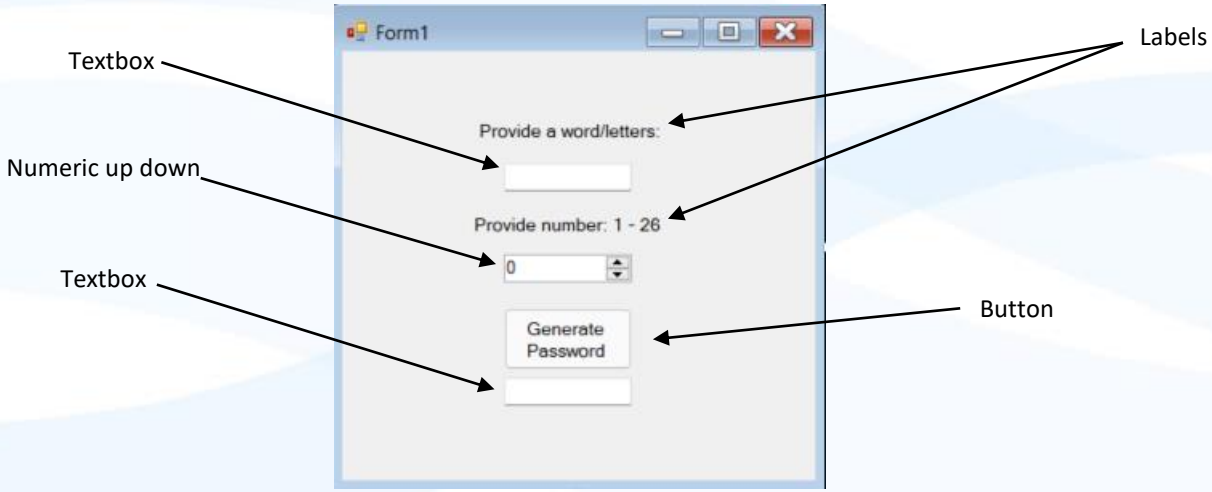
Remember to ask yourself the following questions:

- Inputs and Obtaining Data:
  - Is the user providing us with information, if so, how?
  - Once I have the interface to obtain the data, do I need to manipulate it?
- Processing:
  - How do I solve the problem?
    - » What formulas do I need?
    - » What steps are required?
- Outputs:
  - How are we displaying the information?
  - Do we have to manipulate the data again?

Practical Exercise 5b

The interface:

Now that we have an understanding of the program we would like to implement. We can design the interface.



Practical Exercise 5b

The code:

– Declaring variables:

```
20          //Variables
21
22          string alphabet = "qwertyuiopasdfghjklzxcvbnm",
23                 characters = "!@#$%^&*()_+-={}|[]`,.<>?/",
24                 providedWord, finalPassword;
25
26          char selectedLetter, selectedChar;
27          int indexForAlphabet, providedNumber, randomNumberGenerated, indexForCharacters;
28          Random RandomNumber = new Random();
```

Practical Exercise 5b

The code:

- Get your inputs and deal with data manipulation.

```
37          {
38              // Obtaining inputs
39
40              providedNumber = Convert.ToInt32(nudProvidedNumber.Value);
41              providedWord = txtWord.Text;
```

- Processes.

```
43          // Processes
44
45          randomNumberGenerated = RandomNumber.Next(1, 101);
46
47          indexForAlphabet = RandomNumber.Next(1, providedNumber);
48          selectedLetter = alphabet[indexForAlphabet];
49
50          indexForCharacters = RandomNumber.Next(1, providedNumber);
51          selectedChar = characters[indexForCharacters];
52
53          providedWord = providedWord.ToUpper();
54          finalPassword = providedWord + selectedLetter + randomNumberGenerated + selectedChar;
```

Practical Exercise 5b

The code:

- Output.

```
56          // Output
57
58          txtPassword.Text = Convert.ToString(finalPassword);
```

- Overall structure.

```
30      private void btnGenerate_Click(object sender, EventArgs e)
31      {
32          if(txtWord.Text == "" || nudProvidedNumber.Value == 0)
33          {
34              MessageBox.Show("There are missing values");
35          }
36          else
37          {
38              // Obtaining inputs
39
40              providedNumber = Convert.ToInt32(nudProvidedNumber.Value);
41              providedWord = txtWord.Text;
42
43              // Processes
44
45              randomNumberGenerated = RandomNumber.Next(1, 101);
46
47              indexForAlphabet = RandomNumber.Next(1, providedNumber);
48              selectedLetter = alphabet[indexForAlphabet];
49
50              indexForCharacters = RandomNumber.Next(1, providedNumber);
51              selectedChar = characters[indexForCharacters];
52
53              providedWord = providedWord.ToUpper();
54              finalPassword = providedWord + selectedLetter + randomNumberGenerated + selectedChar;
55
56              // Output
57
58              txtPassword.Text = Convert.ToString(finalPassword);
59          }
60      }
```

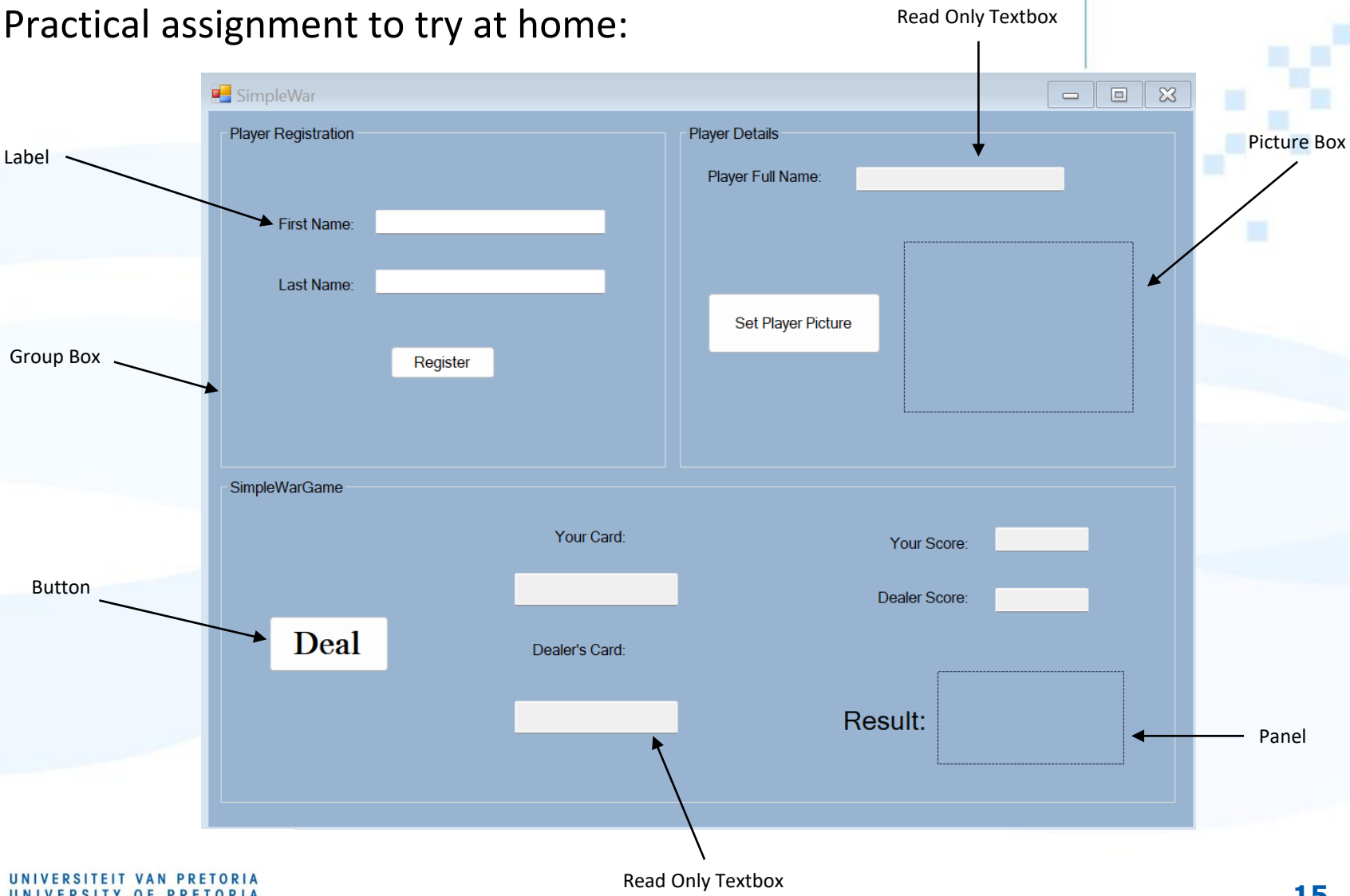
Practical Exercise 5c

Practical assignment to try at home:

- You need to code a program that will allow a user to play a simplified version of **War**, the card game.
- Rules:
  - 1v1 against the computer. Your profile is needed before, with your name and profile photo.
  - A card is drawn for each player, you and the computer.
  - If the computer's card is greater or equal to the player's card, then the computer wins. The computer's score is increased by 1.
  - If the player's card is greater than the computer's card, then the player's score is increase by 1.
- For more in-depth instructions, please refer to the instruction document and rubric.

Practical Exercise 5c

Practical assignment to try at home:



Practical 5 Submission

Submit your Practical 5c project on ClickUP as follows:

- Due Date: 24th April 2023.
- Name your project, INF154Prac5xxxxxxxxx (where xxxxxxxxx is your student number) and compress (zip) your project.
- Submit under the Practical 5 submission link.