Contents

If Statements

This lab serves multiple goals:

- To reinforce your understanding of boolean values,
- To practice conditional statements, and
- (Optional) To solve and implement solutions to simple, concrete problems.

The second part may seem repetitive, but you need to practice `if` statements *a lot* to understand their mechanics and to be able to write them properly.

Basic Conditional Statements

Testing and Improving Conditional Statements

Consider the following code[1]:

```
int yourAge; // This variable will contain the user's age.
Console.WriteLine("Please, enter your age"); // We ask the user.
// We read from the user and convert their answer into an int.
yourAge = int.Parse(Console.ReadLine());

// The rest of the code tests the value of yourAge and
// displays a message based on its value. You are asked
// to explain this code next.

if (yourAge < 0)
{
    Console.WriteLine(
     "I believe you made a mistake, an age cannot be negative!");
}
else if (yourAge > 2000)
{
```

[1]The information about the age of majority comes from wikipedia[2].

```
    Console.WriteLine(
     "I believe you made a mistake, nobody can live that long!");
}
else if (yourAge >= 18)
{
    Console.WriteLine(
       "In all States but Alabama, Nebraska, Mississippi"
     + " and Puerto Rico, you have reached the age of majority.");
}
else if (yourAge >= 19)
{
    Console.WriteLine(
        "In all States but Mississippi and Puerto Rico,"
        + " you have reached the age of majority.");
}
else if (yourAge >= 21)
{
    Console.WriteLine(
     "You have reached the age of majority in all US states.");
}
```

1. *Without executing it,* write down what you expect to be displayed if the user enters
   (a) 29339
   (b) -15
   (c) "That's confidential"
   (d) 10
   (e) 18
   (f) 19
   (g) 22
2. Download a solution containing this code as its `Main` method[3].
3. Execute it, providing the values written above. Were your expectations correct? If not, revise it and make sure you understand the logic of the program.
4. There is at least one issue with this code as "You have reached the age of majority in all US states." will never be displayed. Can you understand why?
5. Fix the program so that all the messages can be displayed when relevant. Feel free to reorder statements or to use conjunction, disjunction, etc. to alter the conditions.

[3] ./code/projects/voting_age.zip

Writing Simple Conditional Statements

Read all the instructions in this part before starting to type code. Create a new project, and write portions of code that perform the following:

1. Ask the user for an integer, and display on the screen "You were born after me" if the number is strictly greater than your year of birth.
2. Ask the user for an integer, and display on the screen "Between -1 and 1" if the number is greater than or equal to -1 and less than or equal to 1.
3. Ask the user for an integer, and display on the screen "Not between -1 and 1" if the number is greater than 1 or less than -1.
4. Ask the user for an integer, and display on the screen "Odd" or "Even", depending if the number is odd or even.
5. Ask the user for an integer, and display on the screen "Negative" if the integer is negative, "Positive" if the integer is positive, and nothing if the integer is 0.
6. Ask the user for an integer, and display on the screen "positive and odd" if the number is positive and odd, "positive and even" if the number is positive and even, "negative and odd" if the number is negative and odd, "negative and even" if the number is negative and even, and "You picked 0" if the number is 0.

For each of those questions, write on paper whenever you should use `if`, `if-else`, `if-else-if`, and what the condition(s) should be. Once you feel confident, write the code in your IDE, and then test it intensively; enter all kinds of values (positive and odd, negative and even, 0, and remember that 0 is even, etc.) and make sure that what is displayed on the screen is always correct.

Observation: How to Construct a Value Progressively

Please, read this part only once you have solved the last question of the previous exercise. You were asked the following:

> Ask the user for an integer, and display on the screen "positive and odd" if the number is positive and odd, "positive and even" if the number is positive and even, "negative and odd" if the number is negative and odd, "negative and even" if the number is negative and even, and "You picked 0" if the number is 0.

A possible answer is:

```
// First, we obtain the number from the user:
int answer;
Console.WriteLine("Enter an integer");
```

```
answer = int.Parse(Console.ReadLine());

// Then we test and display accordingly:
if (answer >= 0 && answer % 2 == 0){
    Console.WriteLine("Positive and even");
}
else if (answer >= 0 && answer % 2 != 0) {
    Console.WriteLine("Positive and odd");
}
else if (answer < 0 && answer % 2 == 0){
    Console.WriteLine("Negative and even");
}
else if (answer < 0 && answer % 2 != 0){
    Console.WriteLine("Negative and odd");
}
```

That is a lot of repetition! And, as you know, it is not good practice to copy-and-paste the very same code, as it requires twice the editing every time you make an update!

An alternative approach is to "progressively" construct the message we will be displaying:

```
// First, we obtain the number from the user:
// (This part is unchanged)
int answer;
Console.WriteLine("Enter an integer");
answer = int.Parse(Console.ReadLine());

// Then, we declare the string that will
// contain the message we will be displaying:
string msg;

// Then, we test the value, constructing
// the string to display as we go:
if (answer >= 0)
{
    msg = "Positive";
}
else // This is the same as "if (answer < 0)".
{
    msg = "Negative";
}
if (answer % 2 == 0)
{
   msg += " and even"; // The "+=" operator adds to the end of the string
}
```

```
else // This is the same as "if (answer % 2 != 0)".
{
    msg += " and odd";
}

// And *finally* we display the message:
Console.WriteLine(msg);
```

This is arguably much better, for the following reasons:

- There is less repetition (e.g., we do not have to repeat `answer >= 0`),
- The conditions are simpler (e.g., no conjunction), and
- Since the two conditions (being odd / even, and being positive / negative) are actually independent, it seems more logical to test them separately.

Pushing Further (Optional)

This part asks you to read and understand a simple problem and to design, implement, and test a solution to it. You are asked to write a simple program that computes the total price for a group of people to enter a park.

Your program should:

- Ask the user how many adults and how many children want to enter the park,
- If the group comprises 6 persons or more, offer to sell a group pass for $\$30$ (that allows everyone in the group to enter the park), and
- Compute and display the total price on the screen, knowing that:
  - Adults pay $\$7$,
  - Children pay $\$4$, and
  - *If* purchasing the group pass allowed the group to save money (which isn't always the case!), you should display on the screen the amount saved.

Some tips:

- When asking "yes" / "no" questions, treat "y" and "Y" as a "Yes", and any other string as a "No".
- Note that we will sell the pass even if the user is not saving any money by doing so (e.g., if 6 children want to enter, $\$4 \times 6 = \$24 < \$30$, but we would still sell them the pass).

Here is an example of execution, where the user input is underlined, and hitting "enter" is represented by ↵:

```
How many adults?
2 ↵
How many children?
4 ↵
Since you are more than 6, you can buy a group pass for
 ↪  $30. Are you interested (please enter "Y" for "yes"
 ↪  or anything else for "no")?
Y ↵
Your total is $30.00 (you saved $0.00).
```