

# Lab 2 – Decisions & Boolean expressions

Topics Covered:

- Boolean expressions
- Decisions: examples & exercises

## Part 1: Pre-Lab Questions

*Question 1:* In the following program, the variable x is set to 8. Which of the following statements will print the message "x is larger than 6"?

```
x = 8
```

- a) if x>6:  
    print("x is larger than 6")
- b) if x<6:  
    print("x is larger than 6")
- c) if x is > 6:  
    print("x is larger than 6")
- d) when x>6:  
    print("x is larger than 6")

*Question 2:* Which of the following statements about if statements is incorrect?

- a) An if statement can have many else-ifs (elif) as needed
- b) Indenting consistently is not only syntactically required in Python, it also makes code much easier to follow
- c) If statements start with an “if” and always end with an “else” clause
- d) If statements allow your code to branch depending on a condition

*Question 3:* To familiarize yourself with Boolean expressions, select the correct expression so that the print statement prints out True.

```
a = 12  
b = 7  
t = True  
c = 'e'
```

- a) print((a > 0) and (b > 0))
- b) print((a > 0) or (b > 0))
- c) print((a > 0) and not (b > 0))
- d) Both a) and b)

*Question 4:* To familiarize yourself with Boolean expressions, select the correct expression so that the print statement prints out True.

```
a = 12  
b = 7  
t = True  
c = 'e'
```

- a) `print('a' > c) and ('E' != c))`
- b) `print('a' > c) or ('E' != c))`**
- c) `print('a' > c) and not ('E' != c))`
- d) Both a) and b)

*Question 5:* To familiarize yourself with Boolean expressions, select the correct expression so that the print statement prints out True.

```
a = 12  
b = 7  
t = True  
c = 'e'
```

- a) `print((a % b != 0) and (b * 2 > a))`
- b) `print((a % b != 0) or (b * 2 > a))`
- c) `print((a % b != 0) and not (b * 2 > a))`
- d) Both a) and b)**

*Question 6:* To familiarize yourself with Boolean expressions, select the correct expression so that the print statement prints out True.

```
a = 12  
b = 7  
t = True  
c = 'e'
```

- a) `print((t or not(a > b)) and t)`
- b) `print((t or not(a > b)) or t)`
- c) `print((t or not(a > b)) and not t)`
- d) Both a) and b)**

## Lab 2 – PART 2

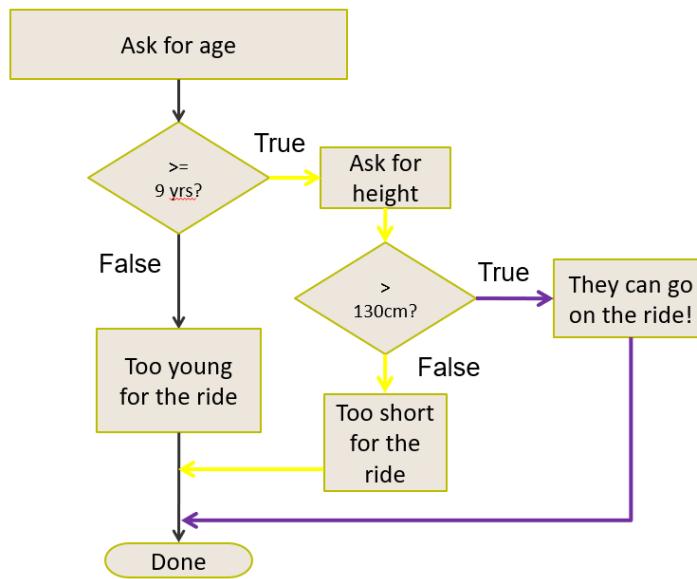
### 1. Follow the instructions below to construct a complete program.

We are going to make a program for an amusement park to determine if a customer can go on a ride. We will go through the process of implementing a nested if statement.

These are the client's specifications:

- Get the client's age
- If they are 9 years old or over
  - Get the customer's height
  - If they are over 130 cm
    - Provide confirmation that they can go on the ride
  - Else
    - Tell them they are too short for the ride.
- Else
  - Tell them they are too young for the ride.

Here is a flowchart based on the client specifications. We will build our code based on this.  
*Explain the flowchart to students.*



Let's code this!

We will start at the beginning. We need to prompt the user for their age. Remember that we have to cast since the input will be string by default.

```
|age = int(input("Enter your age: "))
```

Next in our flowchart, we have a branching condition. If the age is larger or equal to 9, we do something, otherwise we do something else. We will start with the if statement if they are  $\geq 9$ . Remember that with relational operators, the greater than symbol comes before the equal sign.

```
|age = int(input("Enter your age: "))
|if age >= 9:
```

Now we can write what happens if this if statement is true. From our flowchart, if the age is  $\geq 9$ , we ask for their height. We can do this with an input statement, and again we use casting because height is a number and not a string. Modify your code to look like:

```
|age = int(input("Enter your age: "))
|if age >= 9:
|    height = float(input("Enter your height in cm: "))
```

Now from our flowchart, we have another branching condition which is their height. If they are over 130 cm in height, we want to do something. We can add another if statement checking their height. Notice that this is still within our first if statement because we only check the height if they match the condition of being  $\geq 9$  years old. If they are younger than 9, there is no point in checking their height because they are already ineligible to go on the ride, and hence we only prompt for their height if they are  $\geq 9$ . Modify your code as follows:

```
|age = int(input("Enter your age: "))
|if age >= 9:
|    height = float(input("Enter your height in cm: "))
|    if height > 130:
```

Now, if the first if statement is true and they are  $\geq 9$ , and if the second if statement is true and they are over 130cm, then they are eligible for the ride and we can print that message. Modify your code to:

```
|age = int(input("Enter your age: "))
|if age >= 9:
|    height = float(input("Enter your height in cm: "))
|    if height > 130:
|        print("You may go on this ride!")
```

If they are not over 130cm, we want to let them know they are too short for the ride. We make an else clause that corresponds to the if statement that checks for their height.

```
|age = int(input("Enter your age: "))
```

```
if age >= 9:  
    height = float(input("Enter your height in cm: "))  
    if height > 130:  
        print("You may go on this ride!")  
    else:  
        print("You are too short for this ride.")
```

Finally, if they are less than 9 years old and the first if statement is not true, we want to let them know that they are too young for the ride. We do this by making an else clause that corresponds to the first if statement. Your final code should look like:

```
age = int(input("Enter your age: "))  
if age >= 9:  
    height = float(input("Enter your height in cm: "))  
    if height > 130:  
        print("You may go on this ride!")  
    else:  
        print("You are too short for this ride.")  
else:  
    print("You are too young for this ride.")
```

Now, try some different test cases.

## 2. Identify errors and fix the code.

There are 4 different errors in the following code segment; there may be syntax and logic errors. Design some simple test cases to find the logic errors. Find the 4 errors and correct the code to work.

```
1. IDEAL_CREDIT_SCORE = 720  
2.  
3. userScore = int(input("Please enter your credit score: "))  
4. housePrice = int(input("Please enter the price of the house: "))  
5.  
6. if userScore => IDEAL_CREDIT_SCORE:  
7.     downPayment = 0.1 * housePrice  
8. else if userScore < IDEAL_CREDIT_SCORE and userScore > "600":  
9.     downPayment = 0.2 * housePrice  
10. else:  
11.     downPayment = 0.3 * housePrice  
12.  
13. print("Your down payment is: ${}").format(downPayment)
```

## 3. Write the following Python program.

Write a Python program that determines the type of weather-appropriate outfit to wear based on the temperature and weather condition. The program should:

- Prompt the user to input the current temperature in degrees Celsius (as an integer).

2. Prompt the user to input the weather condition (e.g., "sunny", "rainy", "snowy", or "windy").
3. Use conditional statements (if, elif, and else) to determine and print the suggested outfit:
  - o If the temperature is above 25°C and the weather is "sunny":  
"Wear light clothing, sunglasses, and a hat."
  - o If the temperature is between 15°C and 25°C and the weather is "windy":  
"Wear a light jacket and comfortable pants."
  - o If the temperature is between 5°C and 15°C and the weather is "rainy":  
"Wear a raincoat and waterproof boots."
  - o If the temperature is below 5°C and the weather is "snowy":  
"Wear a heavy coat, scarf, gloves, and warm boots."
  - o For any other combination of temperature and weather, print:  
"Dress comfortably and check the forecast for details."

**Example I/O:**

```
Enter the current temperature in degrees Celsius: 25
Enter the current weather condition (sunny, rainy, snowy, windy): windy
Wear a light jacket and comfortable pants.
```

```
Enter the current temperature in degrees Celsius: 5
Enter the current weather condition (sunny, rainy, snowy, windy): sunny
Dress comfortably and check the forecast for details.
```

```
Enter the current temperature in degrees Celsius: 31
Enter the current weather condition (sunny, rainy, snowy, windy): sunny
Wear light clothing, sunglasses, and a hat.
```

```
Enter the current temperature in degrees Celsius: 15
Enter the current weather condition (sunny, rainy, snowy, windy): rainy
Wear a raincoat and waterproof boots.
```