Part I — Questions

1. What is sequential processing?

**Solution.** When the code is executed sequentially, without any branching. It implies that the code is processed in the order in which it is presented in the source code: the statement at line \( n \) will always be executed after the statement at line \( n - 1 \) and before the statement at line \( n + 1 \).

2. What is a decision structure?

**Solution.** A decision structure is a test and one or multiple statement blocks that may or may not be executed based on the outcome of the test. Selection and iteration are two examples of decision structures: in the first one, a statement block can be "skipped over" if the test evaluates to false, in the second one, a statement block can be repeated multiple times, as long as the test evaluates to true.

A decision structure makes it possible to have portions of the code executed conditionally.

3. Decide if the following boolean expressions will evaluate to **true** or **false**:

- \( 3 > 2.0 \land \text{false} \)
- \( (4 \neq 3) \lor \text{false} \)
- \( 'A' == 'b' \land \neg \text{false} \)
- \( (! \text{false}) == (\text{true} \lor 4 == 3) \)

**Solution.**

1. **false**
2. **true**
3. **false**
4. **true**

4. For each of the following boolean expressions, decide if it will evaluate to **true** or **false**:

- \( \text{true} \lor 3 > 4 \)
- \( 'A' == 'C' \lor \neg \text{false} \)
- \( 4 != 3 \land \text{true} \)
- \( (\text{true} \land 4 >= 3) == \text{false} \)

**Solution.**

1. **true**
2. **true**
3. **true**
4. **false**

5. What is the relational operator used to determine whenever two values are different?
6. What is a flag?

Solution. A flag is a boolean variable. Its two possible values are true and false.

7. Give three relational operators, and then two logical operators.

Solution. <=, ==, >, and !, &&

8. What would be displayed on the screen by the following code?

```csharp
if (false)
{
    Console.WriteLine("Hello!");
}
Console.WriteLine("Hi!");
```

Solution. “Hi!”

9. Is there a simpler way to write the expression “over21 == true”, assuming that over21 is a Boolean variable?

Solution. We can simply write over21, which will always evaluate to the same value as over21 == true.

10. Assume that x and y are two int variables that have already been initialized (i.e., declared and assigned), write an if statement that assigns 10 to x if y is (strictly) greater than 5.

Solution. if (y > 5) {x = 10;}

11. In C#, is there a difference between = and ==? Write a statement that use =.

Solution. Yes, one equal sign serves to write assignment operator, and two equal signs serve to compare. An example of statement that uses comparison first and assignment second could be: if (x == 9){x = 12;}

12. Assuming a name string was declared and initialized with a value given by the user, write an if statement that displays “I have the same name!” if name contains your first name.

Solution. if(name=="Clément")Console.WriteLine("I have the same name!");

13. Is the following statement correct, i.e., would it compile, assuming myFlag is a bool variable, and myAge is an initialized int variable?

```csharp
if ( myAge > 20 )
{
    myFlag = true
}
```

Solution. No, the semi-colon should come before the closing brace.
14. Write an if statement that prints “Bonjour!” if the value of the char variable lang is 'f'.

Solution. if (lang == 'f') { Console.WriteLine("Bonjour !"); }

15. For each of the following boolean expressions, decide if it will evaluate to true or false when the boolean variables x, y and z are all set to true:

- x || y && z
- !x || y && z

Do the same when they are all set to false.

Solution. For each expression, we give first the value if all the variables are set to true, then the value if all the variables are set to false.

- true, false
- true, true
- false, false
- true, true

16. Write a boolean expression that evaluates to true if a variable x is between 3 (excluded) and 5 (included).

Solution. x>3 && 5>=x

17. Write an if-else statement that assigns 0.1 to z if y is greater or equal than 0, and that assigns −0.1 to z otherwise.

Solution. if (y >= 0) { z = 0.1; } else { z = -0.1; }

18. Write an if-else statement that displays “It's free for you!” if an int variable age is between 0 and 18, and “It's $5.00.” otherwise.

Solution.

if (age <= 18 && age >= 0) {
    Console.WriteLine("It's free for you!");
} else {
    Console.WriteLine($"It's {5M:C}.");
}

19. What will be displayed on the screen by the following program?

    int x = 3, y = 2, z = 4;
    if (x > y) { z += y; }
    if (x > z) { y -= 4; }
    Console.WriteLine($"x is {x}, y is {y}, and z is {z}.");

Solution. “x is 3, y is 2, and z is 6.”
20. What will be displayed on the screen by the following program?

```csharp
int x = 3, y = 2, z = 4;
if (x >= z) { z += y; } else if (x != y) { z *= y; }
y -= 4;
Console.WriteLine($"x is \{x\}, y is \{y\}, and z is \{z\}.");
```

Solution. “x is 3, y is -2, and z is 8.”

21. (We'll use the 24-hour clock, sometimes called the “military time”). Assuming that an `int` variable `hour` has been initialized, write part of a program that would display on the screen “Good morning” if hours is less than or equal to 12, and “Hello” otherwise.

Solution.

```csharp
if (hours <= 12)
{Console.WriteLine("Good morning!");}
else
{Console.WriteLine("Hello");}
```

22. Assuming that `myString` is a string variable, write a statement that print “Hello, Mélodie!” if the value of `myString` is equal to “Mélodie”, and nothing otherwise.

Solution.

```csharp
if (myString == "Mélodie")
{Console.WriteLine("Hello, Mélodie!");}
```

23. What will be displayed on the screen by the following program?

```csharp
int x = 3, y = 2, z = 4;
if (y >= z) { z += y; }
else if (x != y) {
  if (false) { z -= 3; } else { z += x;}
}
Console.WriteLine($"x is \{x\}, y is \{y\}, and z is \{z\}.");
```

Solution. x is 3, y is 2, and z is 7.

24. Rewrite, if possible, the three following `if-else-if` statements as `switch` statements:

```csharp
if (myLang == 'f') { Console.WriteLine("Vous parlez Français ?"); }
else if (myLang == 'e') { Console.WriteLine("Do you speak English?"); }
else if (myLang == 'd') { Console.WriteLine("Sprechen Sie Deutsch?"); }
else { Console.WriteLine("I don't know your language!"); }
if (myCity == "Augusta") { Console.WriteLine("I also live here!"); }
else if (myCity == "Paris" || myCity == "Boone")
{
  Console.WriteLine("I used to live there!");
}
else
{
  Console.WriteLine("I never lived there.");
}
```
### If you think it is not possible or not feasible, explain why.

**Solution.**

```csharp
switch (myLang)
{
    case 'f':
        Console.WriteLine("Vous parlez Français ?");
        break;
    case 'e':
        Console.WriteLine("Do you speak English?" );
        break;
    case 'd':
        Console.WriteLine("Sprechen Sie Deutsch?" );
        break;
    default:
        Console.WriteLine("I don’t know your language!" );
        break;
}

switch (myCity)
{
    case "Augusta":
        Console.WriteLine("I also live here!" );
        break;
    case "Paris":
    case "Boone":
        Console.WriteLine("I used to live there!" );
        break;
    default:
        Console.WriteLine("I never lived there." );
        break;
}
```

Impossible: can’t write `switch` comparing all the possible `float` values!

### 25. Write a switch statement that sets a `double` discount variable to 0.5 if a string day variable contains "Monday" or "Wednesday", 0.25 if day contains "Saturday", and 0.5 otherwise.

**Solution.**

```csharp
switch (day)
{
    case "Saturday":
        discount = 0.25;
        break;
    default:
        discount = 0.5;
        break;
}
```