college CDI

Fundamental notions of programming (NFP)

Class 9

Retrieving, processing and outputting data

Being able to acquire and process data to generate information is what makes our industry called *information technology*. Programs can get data from different sources such as databases, APIs, files or from the users. The gathered information is then processed as needed in a server and may also be sent to other applications, files, databases or back to the users if needed.

Retrieving data in JavaScript using prompt()

Data can be acquired from users in different ways. For instance, using the JavaScript *prompt()* method allows you to open a client-side window where the user may insert data.

```
<script>
let userName = prompt("Enter your full name:");
</script>
```

Explanation :
In the above example, the method prompt() is used to let the user input data that is stored in a variable named *userName*.

Although available, this method is not the most advisable as some users might find the prompt window annoying and modern browsers even give the option for users to block them. In case the user has opted to block your page from sending prompt windows, your application will most likely break.

Even though prompt windows would work, imagine gathering several informations from a user such as name, address, age, job, etc. The many prompt windows this supposes would make it quite a bad user experience and any correction would be impossible since it is impossible to go back to preceding windows. Therefore, one of the most common ways to get data from users in a JavaScript application consists in using a form.

Processing data gathered from a form

As opposed to the prompt window overlaid over the web page, the form is a part of the page's content which remains accessible to the user. All the information needed and fields are visible so the user knows what's involved. The user may then input the data.

Once the data has been input by the user, the data has to find its way into the program. Just like it was done with the prompt method, it needs to be processed. It can be assigned to variables so different data can be compared or combined in order to generate information.

Outputting results

The input data is processed to generate informations as a result which is usually output. It might not always be the desired result (such as an error message) but it will let the end-user have either the information looked for or information on why the wanted information cannot be output. It is important to remember the output isn't limited to text being displayed on screen. The output can be in the form of changes in the design (colors, sizes, locations, etc), content, media (images, videos, audio), etc.

Programming paradigms

Independently of the languages being used, there are different approaches when it comes to programming, different ways of thinking and considering the development of a solution to solve a problem. That is what we refer to as a paradigm.

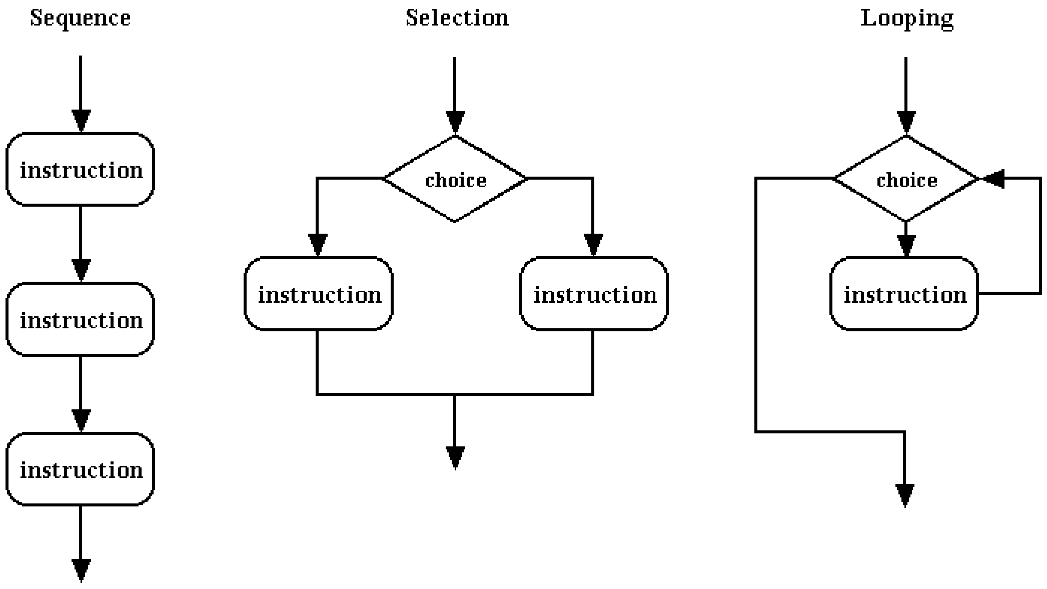
There are different programming techniques (paradigms). Let's start with the basics of ***structured programming*** and ***modular programming*** and a deeper view on the most widely used paradigm : ***object-oriented programming***.

Structured programming

This approach was once widely used, when big applications weren't as complex as they got nowadays, when programmers were basically mathematicians.

In this approach, the application is divided into a basic structure made of several fragments called *building blocks*. These blocks follow a ***sequential structure***, which means the program executes its instructions one after the other in the order they are coded.

In a ***Selection*** or ***conditional structure***, the next module to be executed depends on the result of a series of conditions to be verified within the current module. These conditions are based on *if*, *else*, *elseif*, and *switch cases* or on *repetition* or ***loop structure*** which repeat instructions block as long as a condition is verified (true) such as *while* and *for* structures.

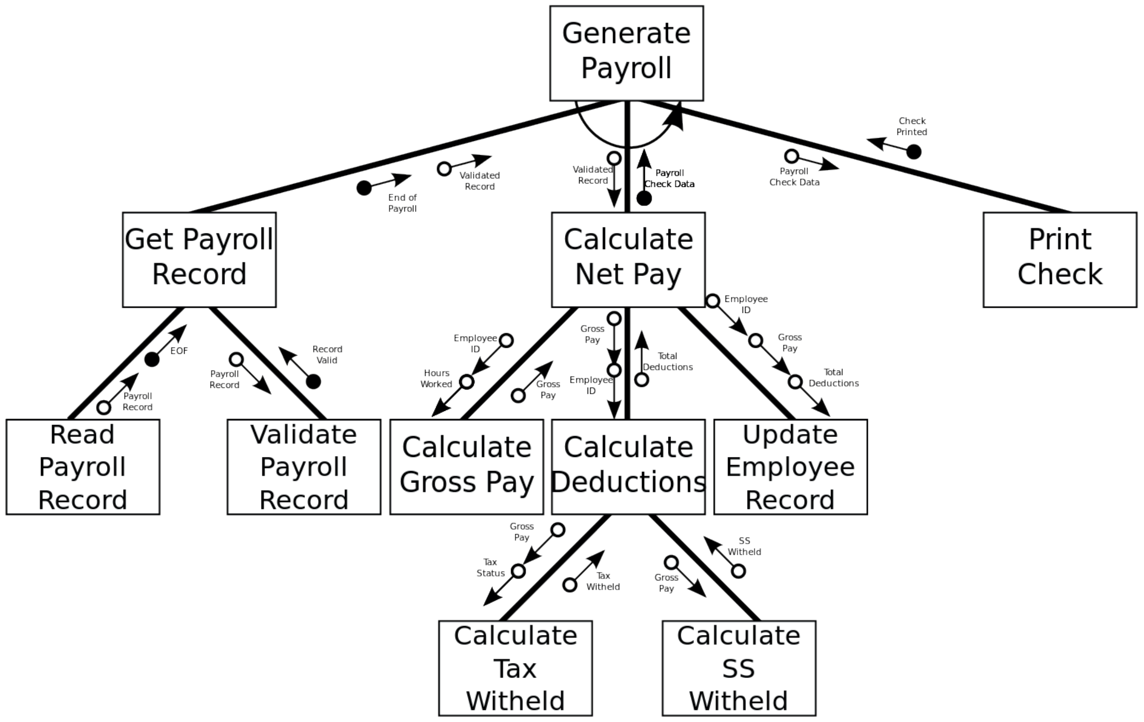


However, this approach to programming has performance issues and require more memory space. In large applications it can become really slow and complex to maintain.

Modular Programming

Modular programming divides the application into modules, each of them comporting their own elements. Somewhat independent, they function as subprograms.

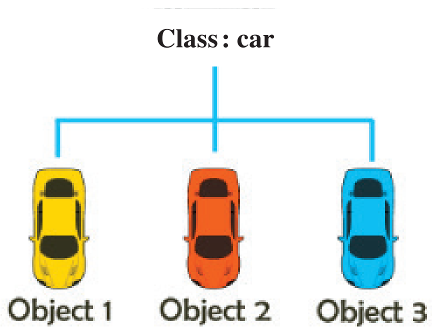
The code being divided into modules executed within the main program, the application becomes better organized, easier to maintain and understand. Modules can also be tested, debugged and expanded individually without a high risk of breaking the entire application.



Object-oriented Programming

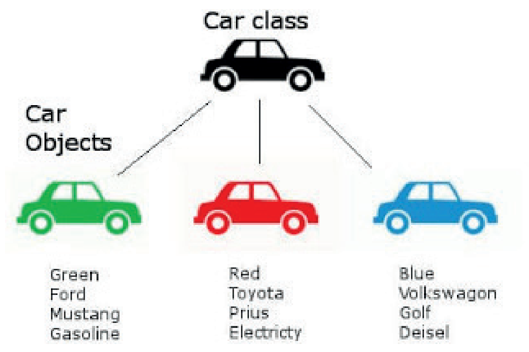
Object-oriented programming (OOP) is the perhaps the most important programming paradigm nowadays. In OOP, an application is considered as a collection of objects communicating with each other. Different concepts first need to be understood : objects, classes, abstraction, encapsulation, polymorphism, and inheritance.

Class
A class consists in a representation of an object's type. Think of it as a template from which objects (instances of the class) are created.



Object
An object is simply... a thing. An object has its own properties and can perform a set of activities related to it.

For instance, a car has different properties such as color, year, engine, and it can also perform a set of actions such as turnOn(), turnOff(), accelerate(), etc.

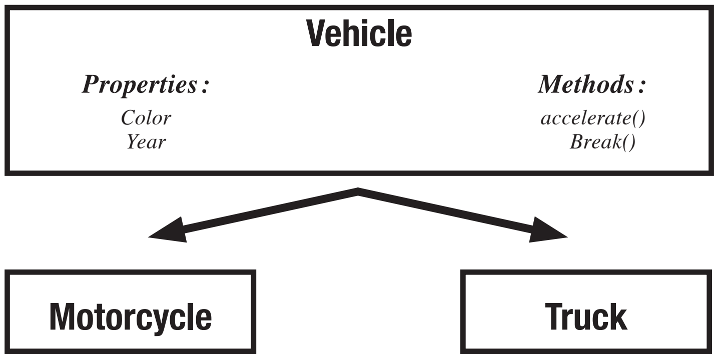


Abstraction
Abstraction signifies that object-oriented programming focuses on the idea, the abstraction of qualities and properties rather than the particularities.

Encapsulation
Encapsulation means that all the resources needed in order for the object to function properly are contained (encapsulated) within the class, the template used to create a given object. It also serves to hide the details of how a class works, not letting part of its logic accessible outside of its scope. A class may only publicly expose the necessary data and offer specific ways for other classes to request information from it.

Polymorphism
Polymorphism simply means many forms. It consists in the ability to process objects differently depending on their data type or class. For instance, a class named *shape* could allow, using different specific methods, to produce objects such as triangle, square and circle.

Inheritance
This means ***child*** class can inherit properties and methods from its ***parent*** class. For instance, different classes and objects such as *motorcycle* and *truck* can share common properties like *color* and *year* as well as common methods such as *accelerate()* or *break()* inherited from a parent class named *vehicle*.



Assignment 7 : Personalized HTML document

Using HTML, CSS and Javascript, create a HTML page which will be using variables to personalize the content.

Using Javascript *prompt()* command, assign variables with user inputs requested upon arrival on the page (e.g. : name, gender, age, etc.).

The user's inputs having been stored in variables, use these so the HTML page seems like it was especially made for the user.