# Object Oriented programming and software engineering – Lab 5

Adam Korytowski – 2025

## 1. Class definitions

Class is a data type that similarly to structures is composed of many data types. By creation of a class, the programmer creates a new data type that can be a model of realistic object.

```
class MyClass
{

};
```

As one can see, class definition is simple. It is important to remember about adding a semicolon after the closing bracket

```
MyClass myObject;

MyClass* myObjectPtr;
myObjectPtr = &myObject;
```

Class elements:

In the class definition shown, one can find a blank space (commented as class body). Inside it the elements of the class are declared. Class elements can be any data:

```
class Processor
{
    int n_cores;
    float clck_sp;
    int n_thr;
    char name[80];
};
```

In order to refer to object elements one has to use the following notation:

– object.element
– pointer->element

To associate value to element *n_cores,* in i5 object it is necessary to use following methods:

```
Processor i5;
Processor* i5Ptr;

i5.n_cores = 4;
i5Ptr = &i5;
i5Ptr->n_cores = 4;
```

2. Access specifiers:

In the above example one could notice notation public. It was so called class label.
Classes can have three types of labels:
– private– this means that elements declared in its range will be only available to use inside of the class that they are declared. This means that only member functions of this class will be able to use them;
– public – this means that elements declared in its range will be available inside the class and beyond it;
– protected – similar to private. It provides members to the classes derived from the class in which it is used. Protected fields are also available for derived (child) classes.

By default if there are no labels inside of the class, members are private.

3. Member functions

The member function is used to perform operations on class data members. It is also the only (though not quite) tool for performing operations on private label components. The function is called for a particular object of the class.

```
class Person
{
public:
    string name;
    int age;
    void introduce();
};

void Person::introduce()
{
    cout << "I am " << name << ", " << age << " years old." << endl;
}
```

4. Encapsulation

It is a mechanism of hiding data and restricting direct access to some components of an object, allowing only controlled access through public methods.

```
class Person
{
    int age;
    string name;

public:
    int getAge() { return age; }
    string getName() { return name; }
};
```

Then having a pointer to our class's object, we can call:

```
cout << person->getAge()
cout << person->getName();
```

4. Rebuild the vending machine software to object oriented program (add classes and objects). The program should have:

- at least 2 classes (2 pts)
- at least 5 methods and their definitions outside of class bodies (2 pts)
- at least 1 enum (1 pt)
- at least 2 additional methods for the sake of encapsulation (getter and setter) (2 pts)