Lab 2

ex 1 && 2

- order of execution of constructors for derived class is from base class up
- without overloading method `printDescription()` for class `Chairman` this method will print description of `Student` class, after overloading only `Chairman` description is going to be printed

```
#include <iostream>
#include <string>
using namespace std;

class Student {
private:
  string description = "A student of the group";
public:
  void printDescription();
  Student();
};

Student::Student() {
  cout << "Creating an object of the class "
          "Student, with description: "
       << endl << " " << description << endl;
}

void Student::printDescription() {
  cout << "description: " << description << endl;
}

class Chairman: public Student {
public:
  Chairman();
  void printDescription();
  string description = "A chairman of the group";
};

Chairman::Chairman()
{
   cout << "Creating an object of the class "
           "Student, with description: "
        << endl << " " << description << endl;
}

void Chairman::printDescription()
{
    cout << "description: " << description << endl;
}

int main() {
```

```
//   Student stud;
  Chairman chair;
//   stud.printDescription();
  chair.printDescription();
  return EXIT_SUCCESS;
}
```

```
output:
Creating an object of the class Student, with description:
 A student of the group
Creating an object of the class Student, with description:
 A chairman of the group
description: A chairman of the group
```

ex 3

```
#include <iostream>
#include <string>
using namespace std;

class Student {
private:
  string name_surname = "NO_NAME";
  unsigned int id_number = 0;
public:
  string description = "A student of the group";
  Student(string name_surname, unsigned int id_number);
  void printDescription();
  void printData() {
    cout << " Method print_data() of the base class" << endl;
    cout << " name surname " << name_surname << endl;
    cout << " id number " << id_number << endl;
  }
  string get_surname();
  int get_id_number();
};

int Student::get_id_number()
{
    return id_number;
}

string Student::get_surname()
{
    return name_surname;
}

class Chairman : public Student {
public:
```

```
  string email = "no@noemail";
  Chairman(string name_surname, unsigned int id_number, string email);
  void printData();
  string description = "A chairman of the group";
};

Chairman::Chairman(string name_surname, unsigned int id_number, string
email)
  : Student(name_surname, id_number)
  , email(email) {
  cout << "Creating an object of the Chairman class named: "
       << description << endl;
}

void Chairman::printData()
{
    std::cout<<"Printing data... \n"<< this->get_surname() << " " << this-
>get_id_number() << " " << email;
}

Student::Student(string name_surname, unsigned int id_number_)
  : name_surname(name_surname)
{
  id_number = id_number_;
  cout << "Creating an object of the Student class named: "
       << description << endl;
}

void Student::printDescription()
{
  cout << "Description: " << description << endl;
}

int main() {
  Student stud("Jan Kowalski", 7);
  stud.printDescription();
  cout << "Data:"
       << stud.get_surname() << " "
       << stud.get_id_number() << endl;
  Chairman chair("Aleksandra Nowak", 999, "mail@nomail.dot");
  chair.printDescription();
  cout << "Data:"
       << chair.get_surname() << " "
       << chair.get_id_number() << endl;
  chair.printData();

  return EXIT_SUCCESS;
}
```

```
Creating an object of the Student class named: A student of the group
Description: A student of the group
```

```
Data:Jan Kowalski 7
Creating an object of the Student class named: A student of the group
Creating an object of the Chairman class named: A chairman of the group
Description: A student of the group
Data:Aleksandra Nowak 999
Printing data...
Aleksandra Nowak 999 mail@nomail.dot
```