Exercise 6:

Submit your solutions (source code) to the questions below by email to your instructor and TA(s) by Monday, November 12th (16:30).

Question 1: Introduction to inheritance (25 points).

In this question, you will check your understanding of the basic principles of inheritance. Create the following files: "Person.h", "Student.h", "Student.cpp" and "test_inheritance.cpp". In the file "Person.h" type the following code (there is nothing to complete here):

```
#ifndef PERSON_H
#define PERSON_H

#include <string>

class Person {
public:
 Person(const std::string& n) : name(n) {}

 void set_name(const std::string& s) {
  name = s;
 }

 std::string get_name() const {
```

```
  return name;
 }

private:
 std::string name;
};

#endif // PERSON_H
```

In the file named "Student.h", write the definition of a class Student that inherits (public inheritance) from Person. A Student is a specialized version of Person with the following additional fields (to be declared private):

- int student_id;
- int year;

Student has also the following additional methods (with public access):

- void set_student_id(int id);
- int get_student_id () const;
- void set_year (int y);
- int get_year () const;

Add these methods in the header "Student.h" and add their implementation in "Student.cpp". Student has only one constructor:

- Student (const std::string& name, int year, int id);

The code for the constructor can go in "Student.h". Finally to test your code, you can type in the file named "test_inheritance.cpp" the following code:

```
// test_inheritance.cpp
#include <iostream>
#include "Person.h"
#include "Student.h"

int main(void) {
  Student s("Tanaka", 2, 12345);
  std::cout << "Student's name: " << s.get_name() << std::endl;
  std::cout << "Student's id: " << s.get_student_id() << std::endl;
  return 0;
}
```

Question 2: Access control and inheritance (25 points).

Step 1:

Add a method print() to the class Student:

```
  void print() const {
   // code to print the name, student id and year of the student
   }
```

Step 2:

Modify the access control of the class Person.h in order to let the print() method of Student access the member data "name" directly without using the method "get_name()" of Person. Make sure that the member data "name" is still inaccessible from methods and functions outside of Person and the classes inheriting from Person. To test step 1 and 2, you can type in "test_access.cpp" the following code:

```
// test_access.cpp
#include <iostream>
#include "Person.h"
#include "Student.h"

int main(void) {
  Student s("Tanaka", 2, 12345);
  s.print();
  return 0;
}
```

Question 3: Substitution principle (25 points).

In this question you will exercise your understanding of the substitution principle. Create a file "test_substitution.cpp" and complete the following code:

```
#include <iostream>
#include "Person.h"
#include "Student.h"
```

```
void print_information (const Person& p)
{
  // complete here by printing the name of p
}

void print_information (const Person* p)
{
  // complete here by printing the name of p
}

int main(void)
{
  Student s1("Tanaka", 2, 12345);

  print_information(s1);
  print_information(&s1);

  return 0;
}
```

Question 4: Inheritance and substitution principle (25 points).

Create files "ForeignStudent.h" and "ForeignStudent.cpp". The class ForeignStudent inherits from Student (public inheritance). In addition to Students' methods and data, it has the additional data (private):

- std::string country;

And the additional methods (public):

- std::string get_country() const;
- void set_country(const std::string& c);

Constructor for a ForeignStudent has the form:

- ForeignStudent(const std::string& name, int year, int student_id, const std::string& country);

Define the class ForeignStudent. Add the methods declarations and data in "ForeignStudent.h" and write the implementation details in "ForeignStudent.cpp". Write the code for the constructor in "ForeignStudent.h". Create a file "test_inheritance.cpp" and type the following code (nothing to complete):

```
#include <iostream>
#include "Person.h"
#include "Student.h"
#include "ForeignStudent.h"

int main(void) {
 ForeignStudent fs("Jack", 2, 1577, "HK");
 std::cout << "Name: " << fs.get_name() << std::endl;
 std::cout << "Country: " << fs.get_country() << std::endl;
```

```
 ForeignStudent fs2("James", 4, 15, "Australia");
 fs2 = fs;
 std::cout << "Name: " << fs2.get_name() << std::endl;
 std::cout << "Country: " << fs2.get_country() << std::endl;

 return 0;
}
```