"Ss. Cyril and Methodius" University in Skopje
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Lesson 6

Inheritance

Object oriented programming

Problem 1

Define a class for tennis player. Each player is represented with his name, surname and if he plays in a league (bool).
From this class derive class for ranked tennis player, that will represent player that plays on international level. For ranked player there is also their rank (integer).

Problem 1

Solution 1/2

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

class TennisPlayer {
protected:
    char name[100];
    bool playsInLeague;
public:
    TennisPlayer(const char *name = "",
                 const bool playsInLeague = false) {
        strcpy(this->name, name);
        this->playsInLeague = playsInLeague;
        cout << "constructor TennisPlayer" << endl;
    }

    friend ostream &operator<<(ostream &out, const TennisPlayer &tp) {
        out << tp.name << endl;
        out << (tp.playsInLeague ? "Plays in League" : "Does not play in league"
             ) << endl;
        return out;
    }
};
```

Problem 1

Solution 2/2

```cpp
class RankedTennisPlayer : public TennisPlayer {
private:
    int rank;
public:
    RankedTennisPlayer(const char *name = "unknown",
                       const bool playsInLeague = false, const int rank = 0) :
        TennisPlayer(name, playsInLeague) {
        cout << "constructor RankedTennisPlayer" << endl;
        this->rank = rank;
    }


    friend ostream &operator<<(ostream &out, const RankedTennisPlayer &rtp) {
        out << rtp.name << endl;
        out << (rtp.playsInLeague ? "Plays in League" : "Does not play in league
             ") << endl;
        out << rtp.rank << endl;
        return out;
    }
};


int main() {
    TennisPlayer tp("Novak Djokovikj");
    RankedTennisPlayer rtp;
    cout << tp;
    cout << rtp;
    return 0;
}
```

Problem 2

Define a class for bank account. Each bank account has a name of the user, number and current balance. Add methods for account preview, adding and withdrawing money.
Then define a class for extended bank account, that will allow the user to get loans from the bank. This class should compute the interest if the user owns the bank money.

Problem 2

Solution 1/3

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

class BankAccount {
protected:
    char name[100];
    long number;
    double balance;
public:
    BankAccount(const char *name = "----", const long number = 0,
                const double balance = 0) {
        strcpy(this->name, name);
        this->number = number;
        this->balance = balance;
    }

    void showBalance() {
        cout << name << endl;
        cout << number << endl;
        cout << balance << endl;
    }

    void addMoney(double amount) {
        balance += amount;
    }

    void withdrawMoney(double amount) {
        if (amount <= balance) {
            balance -= amount;
```

Problem 2

Solution 2/3

```
class BankAccountPlus : public BankAccount {
private:
    double limit;
    static const double INTEREST = 10; // % percent
public:
    BankAccountPlus(const char *name = "----", const long number = 0,
                    const double balance = 0, const double limit = 0) :
        BankAccount(name, number, balance) {
        this->limit = limit;
    }

    void withdrawMoney(double amount) {
        if (amount <= balance + limit) {
            balance -= amount;
        } else {
            cout << "The bank is not giving you that much money..." << endl;
            cout << "Balance: " << balance << endl;
            cout << "Limit: " << limit << endl;
        }
    }

    static void showInterest() {
        cout << INTEREST << endl;
    }


};
```

Problem 2

Solution 3/3

```cpp
int main() {
    BankAccount ba("Pero", 6, 1000000);
    BankAccountPlus bap("Mite", 10, 5000, 1000);
    ba.showBalance();
    ba.addMoney(50000);
    ba.withdrawMoney(600000);
    ba.showBalance();
    bap.showBalance();
    bap.addMoney(500);
    bap.showBalance();
    bap.withdrawMoney(7000);
    bap.showBalance();
    bap.showInterest();
    BankAccountPlus::showInterest();
    return 0;
}
```

Materials and Questions

Lectures, exsercises and announcements
courses.finki.ukim.mk

Source code of all examples and problems
https://github.com/tdelev/SP/tree/master/latex/src

Questions and discussion
forum.finki.ukim.mk