

Solutions

Only the solutions we discussed on the practical will be here. All the other tasks are perfect to practice for the written exam.

Rectangle

```
#include<iostream>

using namespace std;

class Rectangle {
private:
    float a;
    float b;
public:
    Rectangle();
    Rectangle(float a, float b);
    Rectangle(const Rectangle& other);

    float area();
    float perimeter();

    void scale(float n);

    ~Rectangle();
};

Rectangle::Rectangle() {
    this->a = 0;
    this->b = 0;
}

Rectangle::Rectangle(float a, float b) {
    this->a = a;
    this->b = b;
}

Rectangle::Rectangle(const Rectangle& other) {
    this->a = other.a;
    this->b = other.b;
}

float Rectangle::area() {
    return a * b;
}

float Rectangle::perimeter() {
    return 2 * (a + b);
}

void Rectangle::scale(float n) {
    a = a * n;
    b = b * n;
}

Rectangle::~~Rectangle() {
}

int main(void) {
    Rectangle x = Rectangle();
    //Rectangle x;
```

```

//Rectangle x();
Rectangle y = Rectangle(5,4);
//Rectangle y(5,4);
Rectangle z = Rectangle(y);
//Rectangle z(y);

cout << y.area() << endl;
cout << y.perimeter() << endl;

z.scale(2);

cout << y.area() << endl;
cout << y.perimeter() << endl;
cout << z.area() << endl;
cout << z.perimeter() << endl;

return 0;
}

```

String class

(We didn't finish this task.)

```

#include<iostream>

using namespace std;

class String {
private:
    char *str;
    int length;
    int c_string_length(const char* s);
public:
    String();
    String(const char* s);
    String(const String& other);

    //void print();
    //int getLength();

    //String concat(String other);

    //~String();
};

int String::c_string_length(const char* s) {
    int i;
    for(i = 0; s[i] != '\0'; i++) {}
    return i;
}

String::String() {
    str = new char[1];
    str[0] = '\0';
    length = 0;
}

String::String(const char* s) {
    length = c_string_length(s);
    str = new char[length + 1];
    for(int i = 0; s[i] != '\0'; i++) {
        str[i] = s[i];
    }
}

```

```
    str[length] = '\\0';  
}  
  
String::String(const String& other) {  
    this->length = other.length;  
    str = new char[this->length + 1];  
    for(int i = 0; other.str[i] != '\\0'; i++) {  
        this->str[i] = other.str[i];  
    }  
    this->str[this->length] = '\\0';  
}  
  
int main(void) {  
    String a;  
    String b("batman");  
    String c(b);  
    return 0;  
}
```