Programming in C++ I, Rotation 4
Jacobs University Bremen
Dr. Kinga Lipskoch

Course: CH08-320142
November 30th, 2018

Assignment 6 - Operator Overloading, Polymorphism and Abstract Classes

- The problems of this assignment must be solved in C++.
- The TAs are grading solutions to the problems according to the following criteria: https://grader.eecs.jacobs-university.de/courses/320142/2018_2r2/Grading-Criteria-C++.pdf

Problem 6.1 *Choosing a random color* (1 point)

Presence assignment, due by 18:30 h today
Write a function which randomly chooses one color from RED, GREEN, BLUE and YELLOW. Write a program which calls this function 10 times and prints the chosen color on the screen. You can adapt and modify the program below.
The following code snippet simulates a die throw 10 times. Use an array of strings for the colors and then randomly choose a color by its index.

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main()
{
    int die;
    int count = 0;
    int randomNumber;
    // init random number generator
    srand(static_cast<unsigned int>(time(0)));
    while (count < 10) {
        randomNumber = rand();
        die = (randomNumber % 6) + 1;
        cout << count << ": " << die << endl;
        count++;
    }
    return 0;
}
```

Problem 6.2 *Fractions II* (2 points)

Continue with your program for **Problem 5.5** in the following manner. Remember to check the mathematical validity of the parameters (you can assume that the nominator and the denominator are always numbers). Use the suggestions from slide 36 (Lecture 5&6) for choosing to write member methods or friend functions. Also consider the suggestions regarding the return types and parameter types.

- Overload the operators +, − for computing the sum and difference of two fractions.
- Overload the operator = for assigning.
- Overload the operators < and > to compare two fractions.

- In your testing program you should be able to enter two fractions from the keyboard using `cin >>`. Determine the greater fraction and print it on the screen using `cout <<`. Also compute the sum and the difference of the two fractions (storing the result in other objects) and print them on the screen (one per line using the overloaded operator and `cout <<`).

In order to implement the addition and subtraction of two fractions you will have to calculate the lowest common multiple (LCM) of the denominators of the two fractions. The LCM can be computed according to the formula: $\mathtt{LCM(a,b)} = \frac{\mathtt{a \cdot b}}{\mathtt{GCD(a,b)}}$, where GCD is the greatest common divisor.

Operations with fractions

Addition:

$$\frac{a}{b} + \frac{c}{d} = \frac{a \cdot \mathrm{LCM}(b,d)/b + c \cdot \mathrm{LCM}(b,d)/d}{\mathrm{LCM}(b,d)}$$

Subtraction:

$$\frac{a}{b} - \frac{c}{d} = \frac{a \cdot \mathrm{LCM}(b,d)/b - c \cdot \mathrm{LCM}(b,d)/d}{\mathrm{LCM}(b,d)}$$

Multiplication:

$$\frac{a}{b} \cdot \frac{c}{d} = \frac{a \cdot c}{b \cdot d}$$

Division:

$$\frac{a}{b} / \frac{c}{d} = \frac{a \cdot d}{b \cdot c}$$

Problem 6.3 *Polymorphism I* (2 points)

- Download and unzip the archive:
  https://grader.eecs.jacobs-university.de/courses/320142/cpp/a6.zip.
  For compiling the files you can use: 1) compile the files as part of a project or 2) from the terminal using g++.
- Draw (using ASCII characters) a diagram how these classes relate to each other and put this into `testvirtual.cpp` as part of your comments.
- For each numbered point in the file `testvirtual.cpp` add a detailed comment about what is happening in the program.
- Like in `Circle.cpp`, output a message on the screen when the method `calcArea()` is being called in any of the classes.
- Add a method to calculate the perimeter for each class definition.
- Change the test program to additionally print the total perimeter of all objects.
- Also print a message on the screen when the method `calcPerimeter()` is being called.
- Add a `Square` class (consisting of a header file and a cpp file), and add a square object to your test program. Consider the relation of a square to the other classes.

Submit a **zip** file containing all your `.h` and `.cpp` files related to this problem.

Problem 6.4 *Polymorphism II* (1 point)

Change `testvirtual.cpp` such that 20 objects (circles, rings, rectangles, squares) are randomly created at runtime. Their colors (`RED`, `GREEN`, `BLUE`, `YELLOW`) and sizes (between 1 and 50) should also be randomly chosen (see **Problem 6.1**). Compute the total area and the total perimeter of all the objects and print them on the screen.
Submit a **zip** file containing all your `.h` and `.cpp` files related to this problem.

How to submit your solutions

- Your source code should be properly indented and compile with `g++` without any warnings (You can use `g++ -Wall -o program program.cpp`). Insert suitable comments (not on every line . . .) to explain what your program does.
- Please name the programs according to the suggested filenames (they should match the description of the problem) in Grader.
  Each program **must** include a comment on the top like the following:
  ```
  /*
     CH08-320142
     a6_p1.cpp
     Firstname Lastname
     myemail@jacobs-university.de
  */
  ```
- You have to submit your solutions via *Grader* at
  `https://grader.eecs.jacobs-university.de`.
  If there are problems (but **only** then) you can submit the programs by sending mail to
  `k.lipskoch@jacobs-university.de` **with a subject line that begins with CH08-320142**.
  It is important that you do begin your subject with the coursenumber, otherwise I might have problems to identify your submission.
- Please note, that after the deadline it will not be possible to submit any solutions. It is useless to send late solutions by mail, because they will not be accepted.

This assignment is due by Wednesday, December 5th, 10:00 AM.