Driving Test 2010

Friday, 26 March 2pm – 4pm

Using the account details given below, attempt all 3 tasks. Write one C++ file per task. As your working directory you may choose `C:\temp\OOP_your_name`*. Make sure to write in the answers to each question on the* ANSWER SHEET*. Once you have finished, save the 3 files* `TaskN.cpp`, `N=1...3`*, to a directory, e.g. to* `C:\temp\SUBMIT_your_name`*, and contact the invigilator.* **Do not log off!** *Hand in your answer sheet and present your submission directory. The invigilator will then reconnect the network cable so that you can email your solutions to* `rn@ic.ac.uk` *and to yourself. You are then free to leave.*

```
Username  :  Exam09
Password  :  ********
Domain    :  MA215-xx or MA410-xx   (this computer)
```

1. *Fibonacci numbers* [**40 marks**]
The generalized Fibonacci numbers are defined by the recurrence relationship

$$F_n = F_{n-1} + F_{n-2}\,, \quad n \geq 2\,,$$

where $F_0, F_1 \in \mathbb{N}$, with $F_0 + F_1 > 0$, are given initial values. For the classical Fibonacci numbers we set $F_0 = 0$, $F_1 = 1$, which yields e.g. $F_{10} = 55$. It is known that $\lim_{n\to\infty} \frac{F_{n+1}}{F_n} = \varphi := \frac{1+\sqrt{5}}{2}$. The corresponding power series

$$s(x) = \lim_{N\to\infty} s_N(x)\,, \text{ where } s_N(x) = \sum_{k=0}^{N} F_k\, x^k\,,$$

is known to converge for $|x| < \frac{1}{\varphi} \approx 0.618$. Write a program that

(a) given n, computes F_n.
(b) given ε, computes $n_\varepsilon := \min\{n \in \mathbb{N} : |\frac{F_n}{F_{n-1}} - \varphi| < \varepsilon\}$ and F_{n_ε}.
(c) given N and x, computes $s_N(x)$.
(d) given ε and x, computes $N_\varepsilon := \min\{N \in \mathbb{N} : |s_N(x) - s_{N-1}(x)| < \varepsilon\}$ and $s_{N_\varepsilon}(x)$.

To this end, your program should contain the following functions.

(a) `int fibonacci(int n, int F0, int F1)`
(b) `int flimit_phi(double epsilon, int F0, int F1)`
(c) `double s_N(double x, int N, int F0, int F1)`
(d) `int limit_s(double x, double epsilon, int F0, int F1)`

2. *Fractions* [**40 marks**]
Implement a class `fraction` so that all of the statements below are executed correctly. Use only `private` member data and do not use `friend` functions.

```
int a = 4, b = 3;
fraction f(1,123), g(1,321), h(1,3), res;
cout << f - h << endl;
res = a * f - g * b;
res = - res + f * h;
cout << res << endl;
f += g / h;
cout << f << endl;
```

Hint: The following function computes the greatest common divisor (gcd) of two nonnegative numbers: `int gcd(int a, int b) { return ( b == 0 ? a : gcd(b, a % b) ); }`

3. *Gibonacci numbers* [**20 marks**]
The Gibonacci numbers are defined by the recurrence relationship

$$G_n = G_{n-1} + G_{n-2}\,, \quad n \geq 2\,,$$

where $G_0, G_1 \in \mathbb{Q}$, with $G_0+G_1 > 0$, are given initial values. It is known that $\lim_{n\to\infty} \frac{G_{n+1}}{G_n} = \varphi := \frac{1+\sqrt{5}}{2}$. The corresponding power series

$$r(x) = \lim_{N\to\infty} r_N(x)\,, \text{ where } r_N(x) = \sum_{k=0}^{N} G_k\, x^k\,,$$

is known to converge for $|x| < \frac{1}{\varphi} \approx 0.618$. Write a program using **templates** that, in conjunction with your code from 2., can be used to

(a) given n, compute G_n.
(b) given ε, compute $n_\varepsilon := \min\{n \in \mathbb{N} : |\frac{G_n}{G_{n-1}} - \varphi| < \varepsilon\}$ and G_{n_ε}.
(c) given N and x, compute $r_N(x)$.
(d) given ε and x, compute $N_\varepsilon := \min\{N \in \mathbb{N} : |r_N(x) - r_{N-1}(x)| < \varepsilon\}$ and $r_{N_\varepsilon}(x)$.

To this end, your program should contain the following template functions, where `T, T1, T2` are typename placeholders.

(a) `T gibonacci(int n, const T& F0, const T& F1)`
(b) `int glimit_phi(double epsilon, const T& F0, const T& F1)`
(c) `T2 r_N(const T2& x, int N, const T1& F0, const T1& F1)`
(d) `int limit_r(const T2& x, double epsilon, const T1& F0, const T1& F1)`

Of course, when implemented correctly, then the new template functions `gibonacci<int>()`, `glimit_phi<int>()`, `r_N<int, double>()` and `limit_r<int, double>()` correspond one-to-one to the earlier defined functions `fibonacci()`, `flimit_phi()`, `s_N()` and `limit_s()` from 1.

Note: In (b) *it is easiest to overload the operator for* `double = fraction - double`.

Name		CID	

Answer Sheet A

1. On setting $F_0 = 3$, $F_1 = 4$ and $\varepsilon = 10^{-6}$, use your program to compute the following values.

Variable	Values
F_{20}	
n_ε and F_{n_ε}	
$s_{10}(0.1)$	
N_ε, for $x = 0.5$, and $s_{N_\varepsilon}(0.5)$	

2. On replacing the second line of the code in the test with

```
fraction f(1,45), g(1,54), h(1,5), res;
```

what are the outputs of your program?

Line	Output
`cout << f - h << endl;`	
`cout << res << endl;`	
`cout << f << endl;`	

3. On setting $G_0 = \frac{1}{2}$, $G_1 = \frac{1}{3}$, use your program to compute the following values (as fractions!).

Variable	Values
G_{20}	
n_ε, for $\varepsilon = 10^{-6}$, and G_{n_ε}	
$r_5(\frac{3}{8})$	
N_ε, for $\varepsilon = 0.003$, $x = \frac{1}{4}$, and $r_{N_\varepsilon}(\frac{1}{4})$	

Finally, list the output of your program for the following computations.

Variable	Values
`gibonacci<short>(90, 0, 1)`	
`gibonacci<unsigned int>(90, 0, 1)`	
`gibonacci<unsigned long>(90, 0, 1)`	
`gibonacci<float>(90, 0, 1)`	