Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für Theoretische Informatik
Dr. B. Gärtner, Prof. Dr. J. Hromkovič

14. Dezember 2010

Informatik für Mathematiker und Physiker Serie 12 HS 10

URL: http://www.ti.inf.ethz.ch/ew/courses/Info1_10/

Skript-Aufgabe 137 (4 Punkte)

Consider the following family of functions:

```
T foo (S i)
{
  return ++i;
}
```

with T being one of the types `int`, `int&` and `const int&`, and S being one of the types `int`, `const int`, `int&` and `const int&`. This defines 12 different functions, and all of those combinations are syntactically correct.

a) Find the combinations of T and S for which the resulting function definition is semantically valid, meaning, for example, that the constness of variables and references is respected. Semantical correctness also means that the compiler will accept the code, because we have already established syntactical correctness. Explain your answer.

b) Among the combinations found in a), find the combinations of T and S for which the resulting function definition is also valid during runtime, meaning that function calls always have well-defined value and effect; explain your answer.

c) For all combinations found in b), give precise postconditions for the corresponding function `foo`.

Skript-Aufgabe 147 (4 Punkte)

Write a program `random_triangle.cpp` to simulate the following random process graphically. Consider a fixed triangle t and choose an arbitrary vertex of t as a starting point. In each step, choose as a next point the midpoint between the current point and a (uniformly) randomly selected vertex of t.

The simulation at each step draws the current point into a Window. Use the window object `ifm::wio` defined in `<IFM/window>` for graphical output, and choose the triangle with vertices $(0,0)$, $(512,0)$, and $(256,512)$. Use the random number generator `ansic` from `choosing_numbers.cpp`. At begin, the program should read in a seed for the random number generator and the number of simulation steps to perform. For testing purposes, let the simulation run for about 100,000 steps.

Skript-Aufgabe 148 (4 Punkte)

Consider the generator `ansic` used in `choosing_numbers.cpp`. First, recall how the linear congruential method for generating random numbers works. Given a *multiplier* $a \in \mathbb{N}$, an *offset* $c \in \mathbb{N}$, a *modulus* $m \in \mathbb{N}$ and a *seed* $x_0 \in \mathbb{N}$, the sequence $x_1, x_2, \ldots$ of natural numbers defined by the rule

$$x_i = (ax_{i-1} + c) \bmod m, \quad i > 0,$$

is a sequence of (pseudo-)random numbers. The generator `ansic` in the program `choosing_numbers.cpp` uses this method and the following parameters: $a = 1103515245$, $c = 12345$, $m = 2^{31}$, and $x_0 = 12345$.

Since the modulus is $m = 2^{31}$, the internal computations of the generator will certainly overflow if 32 bits are used to represent `unsigned int` values. Despite this, the sequence of pseudorandom numbers computed by the generator is correct and coincides with its mathematical definition. Explain this!

Skript-Aufgabe 149 (4 Punkte)

Find a loaded dice that beats the fair dice in the game of choosing numbers. (This is a theory exercise.)

To recall, here are the rules for one round of choosing numbers: Each of two players independently writes down an integer between 1 and 6. Then the numbers are compared. If they are equal, the game is a draw. If the numbers differ by one, the player with the smaller number gets CHF 2 from the one with the larger number. If the two numbers differ by two or more, the player with the larger number gets CHF 1 from the one with the smaller number.

The exercise asks for a loaded dice which wins against a fair dice on average. The fair dice produces every number between 1 and 6 with the same probability $\frac{1}{6}$.

Die **Aufgaben 150** und **151** aus den Vorlesungsunterlagen sind die **Challenge Aufgaben** und geben jeweils 8 Punkte, wenn sie vollständig gelöst werden.

Abgabe: Bis 21. Dezember 2010, 15.15 Uhr.