Technische Universität München
Fakultät für Informatik
Image Processing, Computer Graphics and Visualization using C++
A. Bigdelou, D.I. Shakir, L. Wang

WS 2010
Exercise Sheet 1
(3 PAGES TOTAL)
Total pts: 40+8
October 26, 2010

VERY IMPORTANT, PLEASE NOTE:

- **(IN EXERCISES WHERE YOU ARE ASKED TO COMMIT CODE INTO THE REPOSITORY) IF THE FILES YOU COMMIT DO NOT MEET THE CRITERIA BELOW (EXCEPT WHERE OTHERWISE MENTIONED), YOU WILL GET NO POINTS FOR THE CORRESPONDING PART OF THE EXERCISE (MARKED WITH AN ASTERISK '*') BELOW.**
- **SVN Commit Criteria:**
  - **Only source code (*.cpp, *.cxx, *.h, Makefile, *.sln, or *.vcproj) belongs in the SVN repository!**
  - **Version-control for binary files like *.exe, *.o, *.obj makes no sense!**
  - **The code committed into the SVN repository should have no compilation or linkage error!**
- **If a question requires a written answer, please either bring your written answer with you for the next session, or send a digital copy of your answer (i.e. .doc, .pdf, .txt, or simply a scan of your handwriting ☺) to shakir@in.tum.de for this exercise.**

1. **(3 pts) Operators in C++**
   What are the values of `c1, c2, f1` and `f2` at the end of the following code snippet? Please explain.

```
int f1=12;
int f2=f1;
int c1=55+ ++f1;
int c2=55+ f2++;
```

2. **(8 pts) Getting used to working with MS Visual Studio and SVN**
   a. Checkout the SVN directory https://camplinux.in.tum.de/svn/trunk/teaching/TeachingWs10CPP/<YourFirstName>/MyFirstVSProject (i.e. replace <YourFirstName> with your first name, all letters small). This is going to be your "playground" for this course.
   b. Open the MS Visual Studio solution file (with the extension .sln).
   c. Build and run the application to see what it prints on the command line.
   d. (4 pts) Modify the code as described within the `main()` function.
   e. Build and run the application again to see the difference.
   f. (4 pts *) Commit all the modified files into the repository.

3. <u>(29 pts) I/O, algorithms</u>

a. (1 pt) Add a new MS Visual Studio project (choose "Win32 Console Application" within the "Visual Studio Installed Templates", leave the settings at the end of the wizard at the default values) called "Ex01_IOAlgorithms" to the existing MS Visual Studio solution from Question 2.

b. (1 pt) Add a header file "functiondefs.h" into the "Header Files" folder of the project. All your functions definitions should go here.

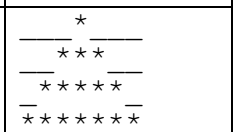
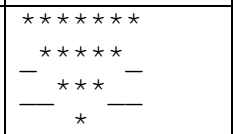
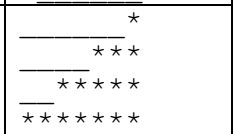
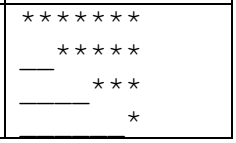
c. (1 pt) Add a .cpp file "functionimpls.cpp" into the "Source Files" folder of the project. All your function implementations should go here.

d. (2 pt) Define an enumeration called `PatternType` with the following values: `topDownEqui, bottomUpEqui, topDownLeft, bottomUpLeft, topDownRight, bottomUpRight`.

e. (8 pts) Define and implement a function
`void getUserInput(unsigned int& lineCount, enum PatternType& pt)`
that:

- ⇒ prompts the user to enter a positive number
- ⇒ parses the user input into an `unsigned int` (hint: use `stringstream` from the STL!)
- ⇒ and keeps asking the user until the input is indeed a positive number (hint: use a while loop!), i.e. when the user enters "-1" (without the quotes), or non-numeric text like "20text" (without the quotes), the function asks again
- ⇒ puts the entered value into the passed input parameter `lineCount`, once the value is entered according to the specifications
- ⇒ prompts the user to enter the desired pattern in the form of an integer (remember: `enum`'s can be used as `int`'s!)
- ⇒ parses the user input into an `unsigned int` (hint: use `stringstream` from the STL!)
- ⇒ and keeps asking the user until the input is within the allowed range (hint: use a while loop!), i.e. one of the values defined in the `enum PatternType`
- ⇒ puts the entered value into the passed input parameter `pt`, once the value is entered according to the specifications

f. (8 pts) Define and implement a function
`void printPattern(unsigned int lineCount, enum PatternType pt)`
that draws the following patterns with the entered `lineCount`, based on the parameter `pt` respectively (hint: use the switch statement):

Value of `pt`	Pattern to draw
`topDownEqui`	
`bottomUpEqui`	
`topDownLeft`	
`bottomUpLeft`	
`topDownRight`	
`bottomUpRight`	

g. (4 pts) Now, your application should prompt the user for input and then draw the corresponding pattern using the described functions within the `main()` function.

h. Save everything within Visual Studio (i.e. the already open solution MyFirstVSProject.sln, and the new header and .cpp file, as well as the file with the `main()` function)

i. Add the new files to the subversion.

j. (4 pts *) Commit all the changes.

4. **(4 pts) [OPTIONAL] Limits of basic datatypes**
What values are in the variables `var1, var2` and `b` at the end of the following code snippet? Can you explain why?

```
int var1 = 2147483648;
int var2 = -2147483649;

unsigned int b=4000;
b=-2;
```

5. **(4 pts) [OPTIONAL]** How are floating-point numbers represented in the computer? Describe it shortly.