Agenda / Learning Objectives:

1. Discuss the answer of quiz 7 question 8 for generating a random 2-digit number that is divisible by 5.
2. Run the following command and extract lab23.tar in your venus account (note the dot):

   cp ~ctse/cs111/lab23.tar . ; tar xvf lab23.tar
3. Review the key tips and pitfalls on using arrays from lab 22 by looking at examples:
   (arrayInitialization.cpp and arrayOutOfRange.cpp)
4. Understand the answers from problem 32 and 36 in prac2Sol.pdf.
5. Go through the key concepts tested in problem 4, 6 and 7 from lab 22.
6. Complete problem 166 from prac2.pdf on using **2D array** in a function.
7. Think about how to approach in answering question 9 from ch.5 of the Absolute C++ textbook. We can first sketch out our ideas with a pseudocode. We will discuss it after MT2. (The solution code is in lab23.tar but let's think about how it can be done first without looking at the answer.)

prac2.pdf on 2D Array: | Solutions
Write a function: 235; 23, 27, 144, 152; 128 136; 148, 156; 160, **164, 166**, 168; 180, **184**, 207
title line: 185e, 197d, 201d, 233d, 237d
code block: 221v
tracing for output: 30, 34; **38, 42**; 117 to 124; 194; 222

Tips (from chapter 5 of Absolute C++ textbook)

Initializing Arrays. In initializing an array,

```
int children[3] = {1, 12, 1};
```

the number of initializers must be no greater than the declared size of the array. There can be fewer initializers. In such an event the first array entries will be initialized from the list, the C++ Standard says the arrays are to be initialized to a zero appropriate to the type. Note that this is true even for primitive types and for class types, where the default constructor is used for the "zero of appropriate type." However, if there are no initializers, and the base type is primitive, the array is not initialized at all. If the base array type is a class type, if there are no initializers, the default constructor is called for each array item.

Pitfalls (from chapter 5 of Absolute C++ textbook)

Array Indices Always Start with Zero. This is a stumbling block for many beginning students that want to number the first spot in an array as 1. However, the first space in an array is actually at index zero.

Array Index Out of Range. The most common programming error made with arrays is an attempt to reference a nonexistent array index. Array indexes most commonly get out of range at the first or last iteration of a loop.

9. TIC-TAC-TOE (from chapter 5 of Absolute C++)

This program is to ask for moves alternately from players X and O. The display should be board-like:

```
1  2  3

4  5  6

7  8  9
```

The players enter their moves by entering a number corresponding to the place to be marked. After each move, the program displays the changed board. After several moves the board may appear:

```
X  X  O

4  5  6

O  8  9
```

This can be done with an array of one dimension to hold the game, a playGame function that calls a play function, which in turn calls a display function and a scoring function.

The play function has a `char` parameter, with the values `'X'` and `'O'` for permissible arguments. The class has private functions `score()` and `display()`, and a `char` array as a data member.

The main function declares the array to hold the game, and initializes the array. The main function then calls `playGame()`. The `playGame()` function alternately calls `play('X',array)` and `play('O',array)`. The function `play` prompts players for input, modifies the data in `array`, calls `displayArray()`, then runs the `score()` member which returns `'X'` if player X wins, `'O'` if player O wins, `'T'` for tie, or `'\0'` (NULL) for no winners. The `score()` function indicates the number of plays remaining.

One thing used here that is not covered in the text but works on all systems having a standard C/C++ library. The library call `int system(char[]str)` function executes any shell command. The argument is any `cstring` that contains the shell command. (A shell command is any command you use to do something at the system's prompt, $, such as `dir`, `ls`, `clear` or `cls`, etc.)

This program uses `system("cls")`, because most Windows based IDEs run programs in a DOS window. The respond to "`cls`" by clearing the screen and placing the cursor in the upper left hand position of the screen. If you are using Linux or some variant of UNIX ™ you should use `system("clear")`, since the UNIX and Linux shell command to clear the screen is `"clear"`. If neither of these works for you, you must read the manual for your system and consult a local expert.

Take care not to allow the overwriting of a position already occupied by an opponent with one of your own. This again points out the necessity for complete testing.