CP2406 Prac-5

By solving the following exercises, you can practice the material discussed in the relevant chapter. Solutions to all exercises are available in the “solutions”-subfolder. However, if you are stuck on an exercise, first reread parts of the chapter to try to find an answer yourself before looking at the solutions.

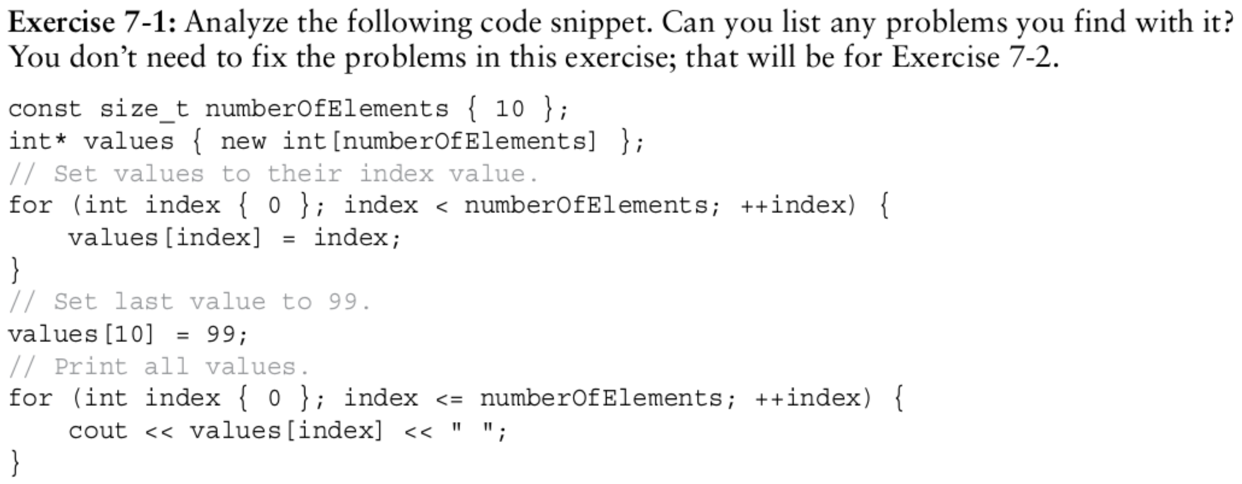
prac05_task1 for Chapter-7-”Memory”

1) Download and unzip the prac’s code.

2) Open the folder “prac05_tasks” in VSCode.

3) NOTE!!! VSCode is getting confused very often. Tested fixes:
*** Close and open it.**
*** Delete “.vscode” folder(s).**
*** Open the Command Palette (Ctrl+Shift+P) and enter Select IntelliSense Configuration. From the dropdown of compilers, select Use g++.exe to configure.**

4) In this task you will complete textbook’s Exercise 7-1, page-246:



Exercise 7-1: Analyze the following code snippet. Can you list any problems you find with it? You don’t need to fix the problems in this exercise; that will be for Exercise 7-2.

```
const size_t numberOfElements { 10 };
int* values { new int[numberOfElements] };
// Set values to their index value.
for (int index { 0 }; index < numberOfElements; ++index) {
    values[index] = index;
}
// Set last value to 99.
values[10] = 99;
// Print all values.
for (int index { 0 }; index <= numberOfElements; ++index) {
    cout << values[index] << " ";
}
```

5) See prac05_task01_ex7p1.cpp

6) Run it to see if it crashes. Hint: it will.

7) Fix the bug to stop it crashing. Hint: There are two major bugs but only one crashes the program.

8) List all the bugs. Hint: there are three bugs.

9) Check your solution against “solutions/prac05_task01_ex7p1_A.cpp”

10) Reflect/Explore: Try printing beyond the array size! It may or may not crash for you. If it does not crash, you have accessed memory that was not yours.

prac05_task2 for Chapter-7-”Memory”

11) Next, let’s complete the textbook’s Exercise 7-2: Rewrite the code snippet from Exercise 7-1 to use **std::array**. (Use #include <array>)

12) Hint: Remember you std::array cannot change its size.

13) Hint: is there a safe way to access all elements in an std::array? Remember array::size()

14) Hint: is there a safe way to reference/set the last element? Google it.

15) Hint: Is there a better loop for read only access? Check chapter-1 pdf.

16) Check your solution against “solutions/prac05_task02_ex7p2_A.cpp”

17) Reflect/Explore: What are the benefits of the read-only loop in the solution?

prac05_task3 for Chapter-7-”Memory”

18) Next, let’s compete the textbook’s Exercise 7-2: Rewrite the code snippet from Exercise 7-1 to use **std::vector**. Use #include <vector> and vector::push_back(someValue)

19) Hint: Try to reuse as much code from std::array-version as possible.

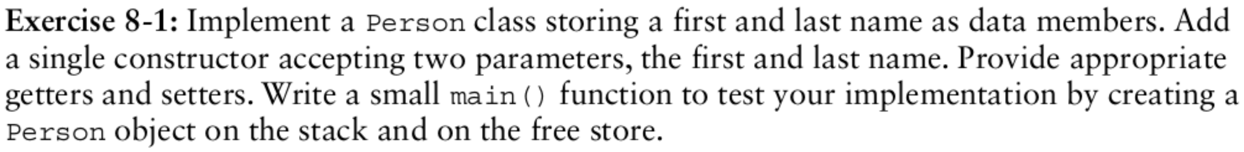
20) Hint: Are you getting Segmentation fault? Check your loading-values loop index.

21) Check your solution against “solutions/prac05_task03_ex7p2_B.cpp”

22) Reflect/Explore: Which one was easier for you to work with? Array or vector? Opinion: array:values[index]=someValue is easier than vector::values.push_back(someValue)

prac05_task4 for Chapter-8-”Classes”

23) In this task you will complete textbook’s Exercise 8-1, page-282:



Exercise 8-1: Implement a `Person` class storing a first and last name as data members. Add a single constructor accepting two parameters, the first and last name. Provide appropriate getters and setters. Write a small `main()` function to test your implementation by creating a `Person` object on the stack and on the free store.

24) Use prac05_task04_ex8p1.cpp to start.

25) Hint: check how to declare a class in Chapter-8 pdf and the comments in the starter code.

26) Check your progress against “solutions/prac05_task04_ex8p1_A.cpp”, which has on the stack and on free-store objects, getters and setters (but no constructors). Note! Not a complete nor perfect solution.

27) Check your progress against “solutions/prac05_task04_ex8p1_B.cpp”, which has getters, setters AND constructor. Note! Not a complete nor perfect solution.

28) Check your final solution against the textbook’s “solutions_textbook/test.cpp” and “Person.cppm”.

29) Correct your solution to be as close as possible to the textbook’s. When fixing your solution, answer why each line was done the way it was. You may need to re-read the chapter’s pdf.

==================== THE END ======================