



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

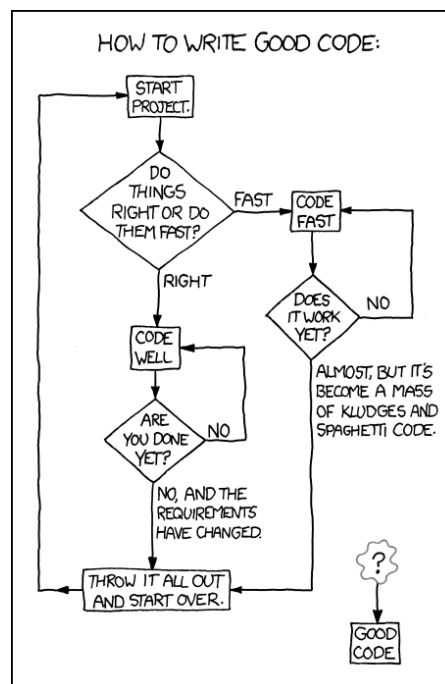
Computer Programming 143

Practical 9

2019

Aim of Practical 9:

- Become familiar with structures
- Become familiar with link lists



Instructions

1. Attendance is **compulsory** for all the practical sessions of your assigned group. See the study guide for more details.
2. The last section (usually the last 30 minutes) of the practical will be used for a test.
3. If more than two tests have been missed for what ever reason, you will receive an **incomplete** for the subject. See the study guide for more details.
4. You must do all assignments **on your own**. Students are encouraged to help each other **understand** the problems and solutions, but each should write his/her own code. By simply copying someone else's code or solutions, you will not build an understanding of the work.
5. You are responsible for your own progress. Ensure that you understand the practical work. Check your work against the memorandum that will be posted on Wednesday afternoons on learn.sun.ac.za.
6. Use H:\CP143 as your Code::Blocks workspace folder for all projects. But it is highly suggested that you also use a **flash drive to backup** all your work.
7. Create a new project **for each assignment**. See *Creating a Code::Blocks Project* in Practical 0 for instructions on how to do this.
8. Include a comment block at the top of each source file according to the format given. It must include the correct filename and date, your name and student number, the copying declaration, and the title of the source file.
9. **Indent your code correctly**. Making your code readable is not beautification, it is a time and life saving habit. Adhere to the standards (refer to the documents on SUNLearn).
10. Comment your code sufficiently well. It is required for you and others to understand what you have done.

Question 9A

Goal: *Add two complex numbers using structures passed to a function.*

Getting started

1. Create a project named Assignment9A. Make sure that this is the active project in the workspace before compiling the program.
2. Include the **standard comment block** above your main function. Also, comment your whole program appropriately.

Program description

1. Create a structure to store the real and imaginary components of a complex number (as floating point numbers). Call the *typedef* of the structure **complex**
2. Create a function with the following prototype:
`complex add_complex(complex n1, complex n2);`
The function must add the two complex numbers and return the result.
3. Ask the user to enter the real and imaginary components of the two complex numbers. See example output below.
4. Add them using the above function and display the result to the screen
5. Example output:

```
For the first complex number
Enter the real part: 2.3
Enter the imaginary part: 4.5

For the second complex number
Enter the real part: 3.4
Enter the imaginary part: 5

Sum = 5.7 + 9.5i
```

6. Ensure that your code is indented correctly and that the `{ }` braces are on the correct lines. Use the prescribed textbook as guideline.
7. Ensure that you copy the **Assignment9A** project folder to a flash drive as a backup.

Question 9B

Goal: *Linked lists.*

Getting started

1. Create a project named Assignment9B. Make sure that this is the active project in the workspace before compiling the program.
2. Include the **standard comment block** above your main function. Also, comment your whole program appropriately.

Program description

1. Essentially, linked lists function as an array that can grow and shrink as needed, from any point in the array. Start by defining a linked list node.
2. In the main function, create a local variable which points to the first item of the list (call it **head**).
3. Set the value to 1 and the next node to NULL.
4. Create a function that adds an item to the end of the list.
5. Create a function that adds an item to the beginning of the list.
6. Create a function that removes a specific item from the list.
7. In the main function, add a node to the beginning of the list with the value 0 and at the end of the list with the value 2 using the above mentioned functions.
8. Remove the middle node.
9. In the main function, iterate over the list and display the values. Your program should display 0 and 2 to the screen.
10. Write a function to delete the list (freeing all the nodes from memory) and delete your linked list.
11. Ensure that your code is indented correctly and that the {} braces are on the correct lines. Use the prescribed textbook as guideline.
12. Ensure that you copy the **Assignment9B** project folder to a flash drive as a backup.

Question 9C

Goal: *Store information in structures.*

Getting started

1. Create a project named Assignment9C. Make sure that this is the active project in the workspace before compiling the program.
2. Include the **standard comment block** above your main function. Also, comment your whole program appropriately.

Program description

1. Create a structure that can store a student's name (a character array of 50 elements), the marks for three tests (integer values out of 100) and the average of the three marks (a floating point number). There are 5 students in the class.
2. Ask the user to insert the names and marks for the 5 students and store it in an array.
3. Create a function that calculates the average of the student's marks and stores the result in the structure (the function should receive the structure **by reference** and not return anything).
4. Calculate the class average of the three tests
5. Display the student's name and average to the screen as well as the class average
6. Example output:

```
Enter the name of student 1:student1
Enter the mark for test 1:10
Enter the mark for test 2:20
Enter the mark for test 3:30
Enter the name of student 2:student2
Enter the mark for test 1:20
Enter the mark for test 2:30
Enter the mark for test 3:40
Enter the name of student 3:student3
Enter the mark for test 1:30
Enter the mark for test 2:40
Enter the mark for test 3:50
Enter the name of student 4:student4
Enter the mark for test 1:40
Enter the mark for test 2:50
Enter the mark for test 3:60
Enter the name of student 5:student5
Enter the mark for test 1:50
Enter the mark for test 2:60
Enter the mark for test 3:70
student1 average for the three tests: 20.000000
student2 average for the three tests: 30.000000
student3 average for the three tests: 40.000000
student4 average for the three tests: 50.000000
student5 average for the three tests: 60.000000

The class average for the three tests is: 40.000000
```

7. Ensure that your code is indented correctly and that the {} braces are on the correct lines. Use the prescribed textbook as guideline.
8. Ensure that you copy the **Assignment9C** project folder to a flash drive as a backup.