GNG1106 Winter 2018 - Assignment 1

Due: Sun. Jan 28, 11:30pm

Instructions

This assignment is to be done INDIVIDUALLY. Use the following instructions to complete and submit this assignment:

- You will need to submit your assignment electronically to Brightspace. Prepare the following:
  - An assignment file in PDF format (this allows you to use your favorite editor to create the PDF file). For question 1, insert the programming models for parts (a) and (b) filled in as per the question instructions. You may fill in the programming model using drawing features of your editor or by hand on paper which is then scanned and inserted into your document (be sure that the hand drawn diagram is legible). For Questions 2 and 3, insert in your assignment file the source code (take care in its appearance), and capture the output from running the program for all test cases. Also submit your **source code** files for questions 2 and 3.
- Place all your files (PDF file and C source code files) in a directory A1_xxxxxxxx where xxxxxxxx is your student number.
- Zip your PDF document and the C source files in a zip file with the name A1_xxxxxx.zip where xxxxxx is your student number.
- Submit the zip file before the assignment deadline via Brightspace. In Brightspace, navigate to the Assignment page and click on "Click to submit Assignment 1" to reach the assignment 1 submission folder. You can also select the Assignment tab to see the Assignment folder pages. The Brightspace video "Assignments" (found in the page https://tlss.uottawa.ca/site/en/connection-to-the-lms#video-tutorials) provides details to help you submit the zip file.
- The questions are provided in both PDF and Word files. You may use the Word file to enter your answers in the document. Be sure to submit a **PDF** file.
- It is NOT permitted to use instructions such as branches and loops that have not yet been covered in the lectures.

➢ Do start the assignment soon and do **not** wait until the last minute. You will be more efficient with a number of smaller efforts over a few weeks before the deadline than one large effort just before the deadline.

Marking Scheme (total 20 marks)

- Question 1: 10 marks
- Question 2: 10 marks
- Question 3: 15 marks

Question 1 (10 marks)

a) (5 marks) The following programming model contains the indicated C program in its code memory. You will be showing how the working memory is used during the execution of this program and how the CPU evaluates the arithmetic expression that assigns a value to the variable **`z`**.

Show the variables created in the working memory and how their values change during the execution of the program.

- Show the values are assigned to the variables. Be sure to show all values that are assigned and replaced. Record successive assignments to variables/parameters as follows:

Variable | ~~7~~ ~~2~~, ~~6~~, ~~4~~, 10 |

- For the operation, show how contents from the working memory are moved to the CPU to calculate the value assigned to variable **z**. Show the results of **all** operations carried out in the CPU, that is, **one line** per operation.
- Finally show the output of the program in the console window.

Code Memory

```
#include <stdio.h>

void main()
{
    double x, y, z;
    x = 50.35;
    y = 28.77;
    z = x*((0.2*y + 1.3)/1.5);
    printf("For x = %5.3f and = %5.3f\n",
            x, y);
    printf("z = %5.3f\n", z);
}
```

Working Memory

CPU

Console

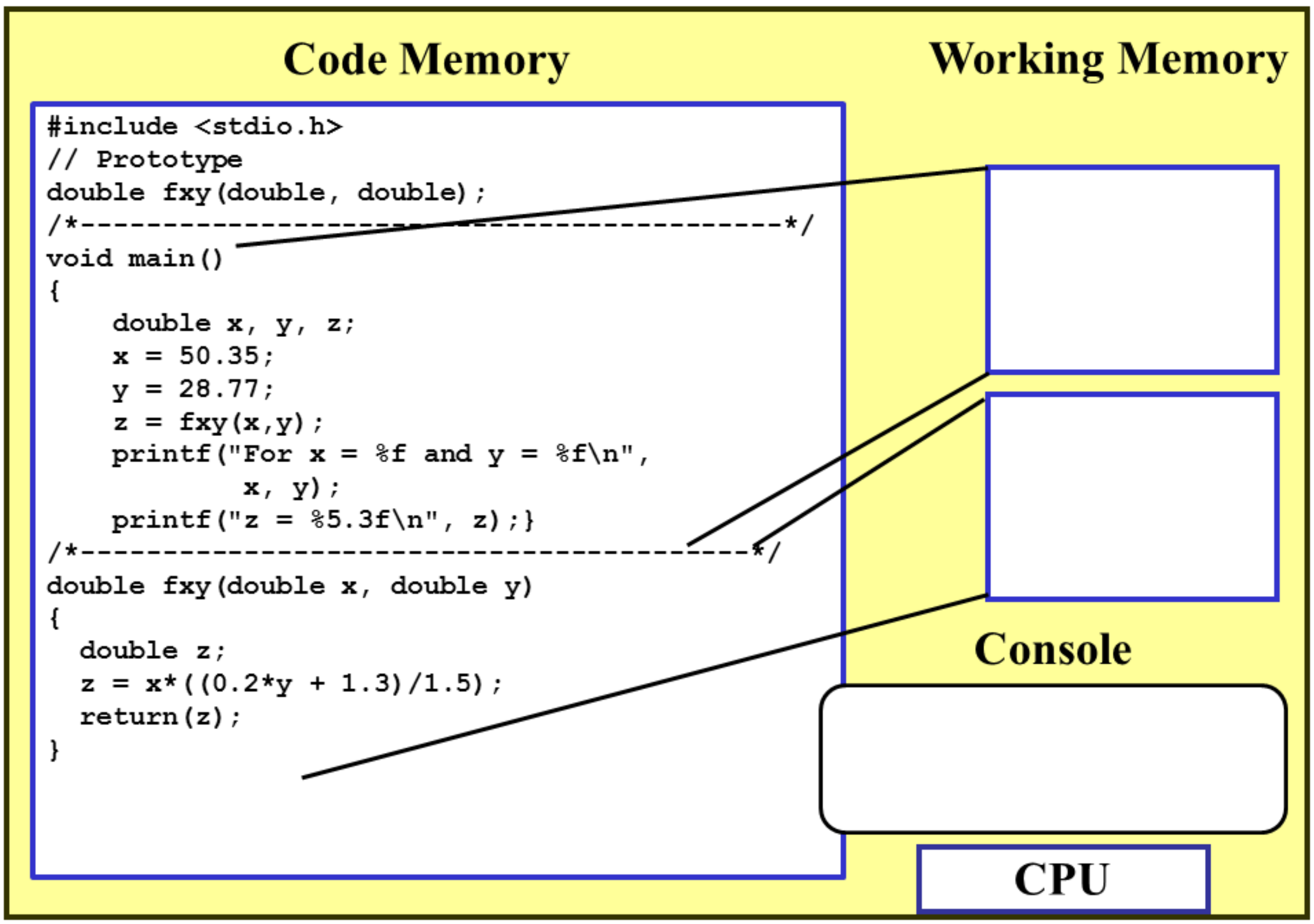
b) (5 marks) The following programming model contains in its code memory the indicated C program composed of 2 functions. You will be showing how the working memory is used during the execution of the two functions. Each piece of working memory is associated to a function using a pair of lines. (Note: the first pair of lines associates the piece of working memory allocated to the function **`main`** and the second pair of lines associates the piece allocated to the function **`fxy`**).

Show how the variables (and parameters) are created in each piece of working memory during the execution of the functions. It is **not** necessary to show how the operations are carried out in the CPU as in the case of part (a).

- Show the values are assigned to the variables. Be sure to show all values that are assigned and replaced. Record successive assignments to variables/parameters as follows:

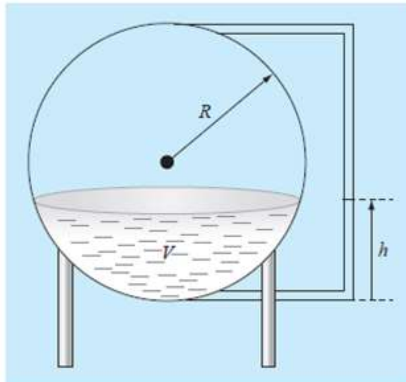
Variable ~~?~~ ~~2~~, ~~6~~, ~~4~~, 10

- Using arrows show how values are copied between the working memory allocated to the function **main** and the working memory allocated to the function **fxy**.
- Finally show the output of the program in the console window.



Question 2 (10 marks)

You are designing a spherical tank to hold water for a small village in a developing country.



The volume of liquid it can hold can be computed as:

$$V = \pi h^2 \frac{(3R - h)}{3}$$

Where V is the volume of the water in m^3;

R is the tank radius in m;

h is the depth of the water.

Develop a program that calculates the volume of the water in the reservoir. The use shall provide the radius *R* of the reservoir, and the depth *h* of the water.

Test your program with the following test cases:

Radius R (m)	Depth h (m)	Water Volume (m^3)
2.00	0.75	3.092505
2.00	2.00	16.755161
5.35	7.45	499.849906
8.10	0.23	1.333400
1.00	2.00	4.188790

Guidelines:

Logic/Strategies

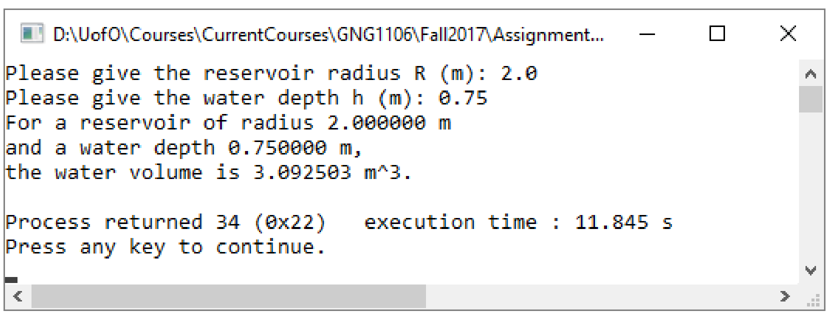
- In the **`main`** function
  - Consider using the following local variables.
    - **`r`**: for storing the radius of the reservoir given by the user,
    - **`h`**: for storing the depth of the water given by the user,
    - **`vol`**: for storing the calculated volume of the water.
  - Prompt the user for the radius of the reservoir and depth of the water. Use short and appropriate prompt messages. Read the values from the keyboard and assign them respectively to the variables r and h.
  - Call a function, say **`calculateVolume`**, to calculate the volume of the water in the reservoir. Store the result returned by the function in the variable **`vol`**. You will need to define this function in your program.

- Display the results with a message of the following form:
  ```
  For a reservoir of radius 2.000000 m
  and a water depth 0.750000 m,
  the water volume is 3.092503 m^3.
  ```
- For the function **`calculateVolume`**
  - Consider using the following parameters
    - **`r`**: gives the radius,
    - **`h`**: gives the depth,
  - Consider using the following local variables
    - **`vol`**: for storing the calculated water volume (note that this variable will contain the value returned).
  - The function calculates the volume of the water in the reservoir (see the equation given at the beginning of the question), stores it in the variable **`vol`** and returns the value stored in **`vol`**.

Test the program using the values provided in the table above. In your assignment file, give the output of your program for all test cases.

The answer to this question should provide:

1) The source code to your program (also insert the source code into the assignment file).
2) The output showing the results of all the test cases; insert the output into the assignment file. The following is an example of the output for the first test case.



Question 3 (15 marks)

The Stefan-Boltzmann law can be employed to estimate the rate of radiation of energy *H* from a surface, as in

$$H = Ae\sigma T^4$$

where *H* is the radiation in watts,
A is the surface area *(*m^2*)*,
e = the emissivity that characterizes the emitting properties of the surface (dimensionless),
σ = a universal constant called the Stefan-Boltzmann constant *(*= 5.67 × 10−8 Wm−2 K−4*)*,
and *T* = absolute temperature (degrees Kelvin).

Develop a program that calculates the rate H for a rectangular surface. The user provides the following data: the length and width of the surface (in meters), the surface emissivity, and the surface temperature in degrees Centigrade. Note that the program must convert the temperature to Kelvin to use the above equation.

Complete this question as follows:

1) First develop a set of test cases (Excel is practical spreadsheet software that allows you to create test cases). Be sure that test cases cover wide ranges for input data (small and large surface areas, low and high values of temperature, etc.). Provide at least 5 test cases. Vary the emissivity between 0.01 and 0.98 and the temperature between -100 and 100 ^{o}C (see http://support.fluke.com/find-sales/Download/Asset/3038318_6251_ENG_A_W.PDFfor the emissivity of various materials).
2) Develop your program using the GNG1106 D template (GNG1106template.c), that is, your program will contain a `main` function and a function that calculates the area of the washer. The `main` function contains instructions to get data from the user, calls the function to compute the total washer area, and displays the results to the user. Be sure to document well your program and follow programming conventions.
3) A few hints:
   a) The value 5.65 x 10^{-8} is represented in a C program with `5.65e-8`.
   b) The calculate tempK^4 (where `tempK` is the variable containing the surface temperature in Kelvin) can be calculated with the expression `tempK*tempK*tempK*tempK`.You can also use the standard C function `pow()`, that is, `pow(tempK,4)`. To use the function `pow`, you must include the header file `math.h`.
4) Include in your assignment report, a table of your test cases, your source code, and the captures output for all your test cases. Also submit your source code file.