

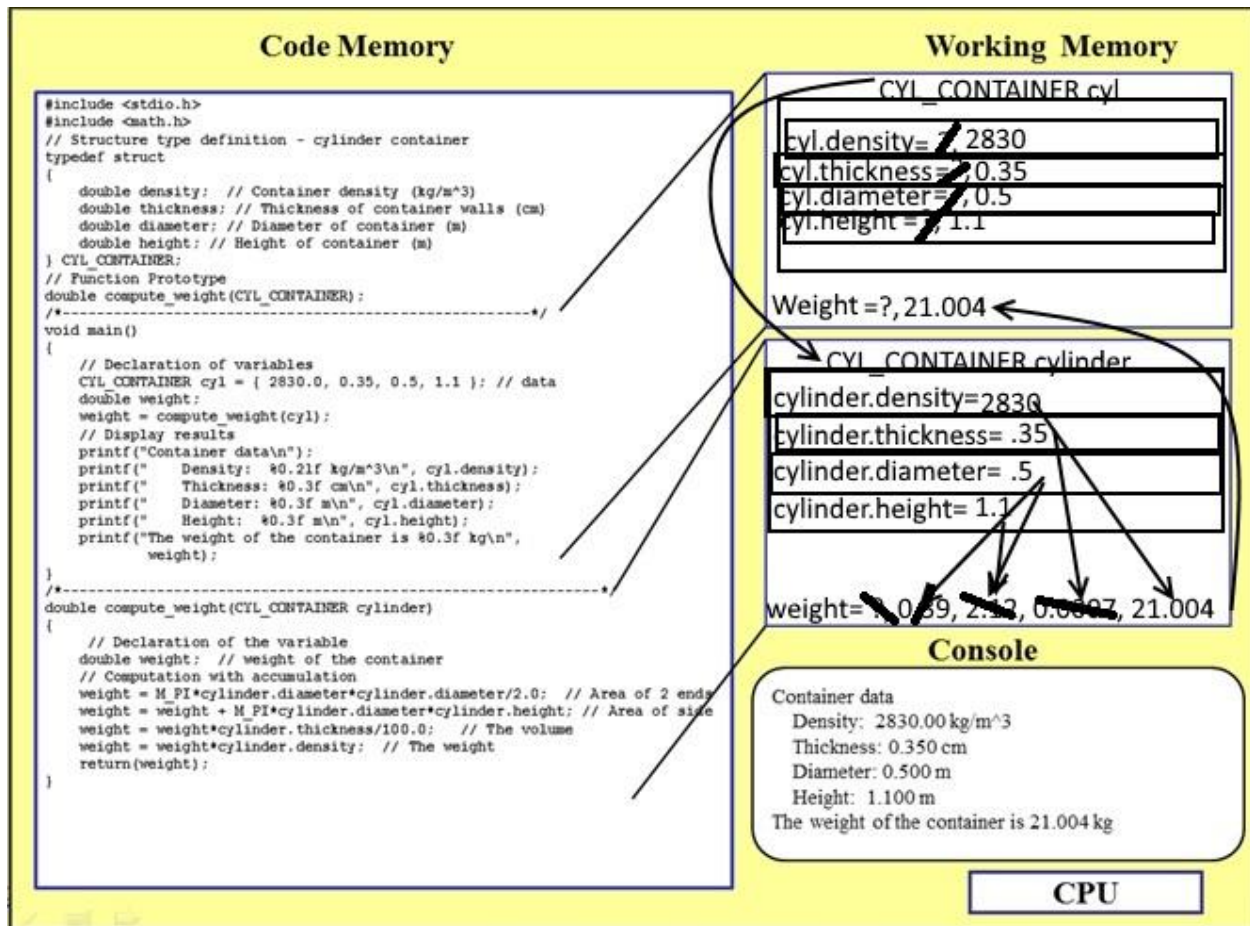
Amit Nayak

300066780

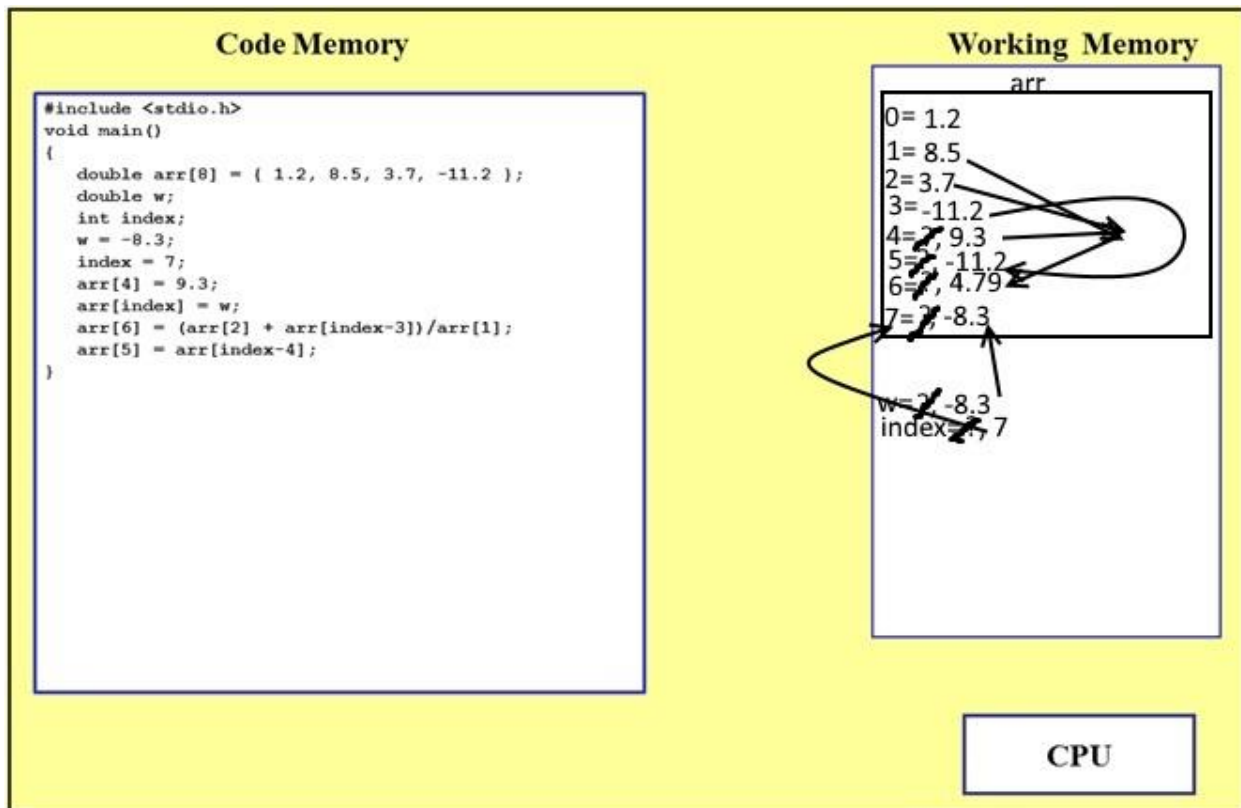
GNG1106 Assignment 4

October 14, 2018

Question 1a:



Question 1b:



Question 2:

Source code:

```
/*
Amit Nayak 300066780 GNG1106 Assignment 4
October 13, 2018
Average Velocity of Water Program:

This program takes in channel characteristics from the user to calculate and return the
average velocity of water.

It returns 25 incremental velocities from the max depth given.

It prints the depths and velocities in a table for the user to see.
*/

#include <stdio.h>
#include <math.h>

// Define symbolic constant
#define D_IX 0
#define V_IX 1
#define LINES 25

typedef struct
{
    char name[15];
    double n;
    double slope;
    double width;
    double maxDepth;
} CHANNEL;

// Prototypes
double getPositiveValue();
int displayTable(CHANNEL);
double computeVelocity(CHANNEL, double);

/*-----
Function: main
Description: Gets from the user values for the equation.
-----*/
```

```

int main()
{
    CHANNEL chan; //structure variable for CHANNEL.

    //This section of code gets all of the user data to fill all the members of CHANNEL
struct.
    printf("Please give the name of the channel: ");
    fgets(chan.name, 15, stdin);
    printf("Please enter the coefficient of roughness: ");
    chan.n = getPositiveValue();
    printf("Please enter the slope: ");
    chan.slope = getPositiveValue();
    printf("Please enter the channel width: ");
    chan.width = getPositiveValue();
    printf("Please enter the maximum depth of the channel: ");
    chan.maxDepth = getPositiveValue();

    //Calls the displayTable function to calculate and print the variables.
    displayTable(chan);

    return 0;
}

/*-----
Function: getPositiveValue
Returns: A value strictly positive (>0)
Description: Reads a real value from the user, checks that it is strictly
             positive, and returns the value.
-----*/

double getPositiveValue()
{
    double value;
    do
    {
        scanf("%lf", &value);
        if(value <= 0.0)
            printf("The value must be greater than zero: \n");
    }
}

```

```

        while(value <= 0.0);

        return(value);
    }

/*-----
Function: displayTable
Description: calls the computeVelocity function to calculate all the
            velocities at each increment of depth. Then it prints the values in a table
            with the the channel characteristics above.
-----*/

int displayTable(CHANNEL chan)
{
    //Initialization of variables.

    double depth = 0; //Depth variable for each calculation of each increment.

    float inc; //Increment variable

    double points[2][LINES]; //LINES value points of depth/velocity to fill the arrays.
    // Note in the 2D array
    //     points[D_IX] is a 1D array that contains the depth values
    //     points[V_IX] is a 1D array that contains the volume values

    inc = chan.maxDepth/(LINES); //Increment based on max depth and number of lines.
    depth = inc; //set depth to first increment

    //runs the calculation for each increment of depth.
    //calls the computeVelocity function for each depth increment.
    for(int i = 0; i < LINES; i++)
    {
        points[D_IX][i] = depth; //set array points to depth value
        points[V_IX][i] = computeVelocity(chan, depth); //set array points to velocity
value

        depth = depth + inc; //increment depth

    }

    //This section of code prints out the table.
    printf("\nChannel Data for: %s", chan.name);
    printf("Coefficient of Roughness: %0.4f\n", chan.n);
    printf("Slope: %0.5f\n", chan.slope);
    printf("Width: %10.2f\n", chan.width);

```

```

printf("Maximum Depth: %0.2f\n", chan.maxDepth);

printf("%10s    %10s\n", "Depth", "Velocity");

printf("-----\n");

for(int i = 0; i < LINES; i++)

    printf("%10.4f    %10.6f\n", points[D_IX][i], points[V_IX][i]); //print depth
and velocity using array values.

}

/*-----

Function: computeVelocity
Parameters: CHANNEL chan, depth.
Description: Passes in the CHANNEL chan members plus the depth.
Calculates using a series of step the average velocity. returns the value.
-----*/

double computeVelocity(CHANNEL chan, double depth)
{
    double u = 0; //Value used for calculation.
    double temp = 0; //Temporary value used in calculation.

    //series of steps for final equation given in the assignment instructions.
    u = sqrt(chan.slope);
    u = u / chan.n;
    temp = ((chan.width/ depth)/(chan.width + (2 * depth)));
    temp = pow(temp, (2.0/3.0));
    u = u * temp;

    return u;
}

```

Outputs:

```
Please give the name of the channel: Channel 3
Please enter the coefficient of roughness: 0.17
Please enter the slope: 0.041
Please enter the channel width: 40
Please enter the maximum depth of the channel: 1.5
```

```
Channel Data for: Channel 3
Coefficient of Roughness: 0.1700
Slope: 0.04100
Width: 40.00
Maximum Depth: 1.50
```

Depth	Velocity
0.0600	7.756067
0.1200	4.876298
0.1800	3.713932
0.2400	3.059721
0.3000	2.631590
0.3600	2.325820
0.4200	2.094561
0.4800	1.912415
0.5400	1.764548
0.6000	1.641663
0.6600	1.537612
0.7200	1.448154
0.7800	1.370260
0.8400	1.301702
0.9000	1.240806
0.9600	1.186282
1.0200	1.137124
1.0800	1.092530
1.1400	1.051856
1.2000	1.014577
1.2600	0.980258
1.3200	0.948540
1.3800	0.919119
1.4400	0.891740
1.5000	0.866183

```
Process returned 0 (0x0) execution time : 30.444 s
Press any key to continue.
```

```
Please give the name of the channel: Channel 2
Please enter the coefficient of roughness: 0.0013
Please enter the slope: 0.0032
Please enter the channel width: 2
Please enter the maximum depth of the channel: 11.5
```

```
Channel Data for: Channel 2
Coefficient of Roughness: 0.0013
Slope: 0.00320
Width: 2.00
Maximum Depth: 11.50
```

Depth	Velocity
0.4600	56.740240
0.9200	29.778690
1.3800	19.693713
1.8400	14.450211
2.3000	11.266883
2.7600	9.146052
3.2200	7.641631
3.6800	6.524830
4.1400	5.666601
4.6000	4.988857
5.0600	4.441703
5.5200	3.991847
5.9800	3.616268
6.4400	3.298570
6.9000	3.026780
7.3600	2.791958
7.8200	2.587308
8.2800	2.407573
8.7400	2.248630
9.2000	2.107203
9.6600	1.980655
10.1200	1.866844
10.5800	1.764014
11.0400	1.670711
11.5000	1.585720

```
Process returned 0 (0x0) execution time : 25.897 s
Press any key to continue.
```

```
Please give the name of the channel: Channel 1
Please enter the coefficient of roughness: 0.035
Please enter the slope: 0.0001
Please enter the channel width: 10
Please enter the maximum depth of the channel: 4.2
```

```
Channel Data for: Channel 1
Coefficient of Roughness: 0.0350
Slope: 0.00010
Width: 10.00
Maximum Depth: 4.20
```

Depth	Velocity
0.1680	0.917961
0.3360	0.566077
0.5040	0.423161
0.6720	0.342380
0.8400	0.289368
1.0080	0.251450
1.1760	0.222759
1.3440	0.200172
1.5120	0.181859
1.6800	0.166669
1.8480	0.153840
2.0160	0.142843
2.1840	0.133301
2.3520	0.124935
2.5200	0.117535
2.6880	0.110939
2.8560	0.105020
3.0240	0.099677
3.1920	0.094829
3.3600	0.090410
3.5280	0.086363
3.6960	0.082644
3.8640	0.079214
4.0320	0.076040
4.2000	0.073095

```
Process returned 0 (0x0) execution time : 17.190 s
Press any key to continue.
```