Lab 9

1. **Looking into programs you developed in previous weeks (Week 1 to Week 8), which programs should be written in multiple modules? Name three and explain why they need to be rewritten into multiple source files.**

*ANS>*

Lab week 3 question number 1, Lab week 7 question number 2 and 3, Lab week 5 question number 1 should be written in multiple modules.

Here are few reasons behind the concept of rewriting into multiple source files.

* Easier to navigate than scroll through one huge file
* Make recompile works only on files related to the change
* Various parts of the program can be programmed by different people
* Code from some files can be put into libraries for future reuse
* At compilation time an error will indicate in what file the problem is (easier to find)
* At compilation time compiler needs much less memory (less requirement to hardware)
* At compilation time easier for compiler to analyze the code (faster).

1. **Try to rewrite your program in Week 4 Question 2 into multiple files, e.g. more than one .c file and .h header.**

*ANS>*

* **main.c**

#include <stdio.h>

#include <string.h>

#include "declaration.h"

#include "pushchar.h"

#include "popchar.h"

#include "header.h"

#define MAX 100

/\*function definition of isEmpty\*/

int isEmpty()

{

if(top==-1)

return 1;

else

return 0;

}

/\*function definition of isFull\*/

int isFull()

{

if(top==MAX-1)

return 1;

else

return 0;

}

* **declaration.c**

#include <stdio.h>

#include <string.h>

int top=-1;

int item;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*string declaration\*/

char stack\_string[MAX];

/\*function to push character (item)\*/

void pushChar(char item);

/\*function to pop character (item)\*/

char popChar(void);

/\*function to check stack is empty or not\*/

int isEmpty(void);

/\*function to check stack is full or not\*/

int isFull(void);

* **header.h**

#include <stdio.h>

#include <string.h>

#include "pushchar.c"

#include "popchar.c"

int main()

{

char str[MAX];

int i;

printf("Input a string: ");

scanf("%[^\n]s",str); /\*read string with spaces\*/

/\*gets(str);-can be used to read string with spaces\*/

for(i=0;i<strlen(str);i++)

pushChar(str[i]);

for(i=0;i<strlen(str);i++)

str[i]=popChar();

printf("Reversed String is: %s\n",str);

return 0;

}

* **popchar.c**

#include <stdio.h>

#include <string.h>

char popChar()

{

/\*check for empty\*/

if(isEmpty())

{

printf("\nStack is EMPTY!!!\n");

return 0;

}

/\*pop item and decrease top\*/

item = stack\_string[top];

top=top-1;

return item;

}

* **pushchar.c**

#include <stdio.h>

#include <string.h>

void pushChar(char item)

{

/\*check for full\*/

if(isFull())

{

printf("\nStack is FULL !!!\n");

return;

}

/\*increase top and push item in stack\*/

top=top+1;

stack\_string[top]=item;

}

1. **Write a complete program that prompt the user for the radius of a sphere and calculates and prints its volume. Use an inline function sphereVolume that returns the result of the volume.**

**Code:**

#include <iostream>

#include <cmath>

using namespace std;

inline double sphereVolume (double radius)

{

return (4.0 / 3.0 \* 3.14159 \* pow(radius, 3));

}

int main()

{

double radius;

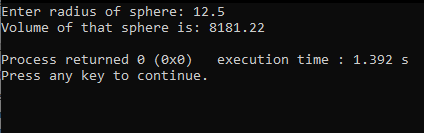
cout << "Enter radius of sphere: ";

cin >> radius;

cout << "Volume of that sphere is: " << sphereVolume(radius) << endl;

}

**Output**:



**4. Write a complete C++ program with the two alternate functions specified below.**

These functions will double the variable count defined in main.

a. Function doubleByValue that passes a copy of count by value, double the copy and returns the new value.

b. Function doubleByReference that passes count by reference via a reference parameter, double the original value of count through its alias, i.e. the reference parameter.

**Code:**

#include <iostream>

using namespace std;

int doubleByValue(int num);

void doubleByReference(int &num);

int main()

{

int num = 5;

int result\_by\_val = doubleByValue(num);

cout << "Return pass by value: " << result\_by\_val << endl;

int result\_by\_ref = num;

doubleByReference(result\_by\_ref);

cout << "Return pass by ref: " << result\_by\_ref << endl;

return 0;

}

int doubleByValue(int num)

{

return num\*2;

}

void doubleByReference(int &num)

{

num = num\*2;

}

**Output:**

