In Class Assignments: Week 7

1.Write an object-oriented C++ program to create a class for room and three objects to calculate the area and volume of three rooms. Use public access specifier and methods for calculations.

#include <iostream>  
using namespace std;  
class room{  
public:  
 float length, width, height;  
 void set\_var(float LENGTH, float WIDTH, float HEIGHT){  
 length = LENGTH;  
 width = WIDTH;  
 height = HEIGHT;  
 }  
 //Object for calculating Volume  
 //not passing parameter  
 float vol\_noPass(){  
 return length \* width \* height;  
 }  
 //Object for calculating Area  
 float area\_noPass(){  
 return length \* width;  
 }  
};  
  
int main() {  
 float l, w, h, volume, area;  
 room c{};  
  
 cout << "Enter Length, Width, Height Values: " << endl;  
 //set variables  
 cin >> l >> w >> h;  
 c.set\_var(l, w,h);  
  
 volume = c.vol\_noPass();  
 area = c.area\_noPass();  
 cout << "No-Passing Parameter Volume: " << volume << " units" << "\nNo-Passing Parameter Area: " << area << " units"<< endl;  
 return 0;  
}

Enter Length, Width, Height Values:

3

6

9

No-Passing Parameter Volume: 162 units

No-Passing Parameter Area: 162 units

Process finished with exit code 0

2. Write a program to find the volume of a rectangle using constructor (with and without passing parameters).

#include <iostream>  
using namespace std;  
  
class room{  
private:  
 float length, width, height;  
public:  
 /\*void set\_var(float LENGTH, float WIDTH, float HEIGHT){  
 length = LENGTH;  
 width = WIDTH;  
 height = HEIGHT;  
 }\*/  
 //not passing parameter  
 float vol\_noPass(){  
 cout << "Enter Length, Width, Height Values: " << endl;  
 cin >> length >> width >> height;  
 return length \* width \* height;  
 }  
 //passing parameter  
 void vol\_pass(float length,float width,float height){  
 cout << "Passing Parameter Volume: " << length \* width \* height << " units" << endl;  
 }  
};  
  
int main() {  
 float l{5}, w{10}, h{3}, volume;  
 room c{};  
  
 /\*//set length, width, height variables from main  
 c.set\_var(l, w,h);\*/  
  
 //find volume with passing a parameter  
 c.vol\_pass(5, 10, 3);  
 //find volume without passing a parameter  
 volume = c.vol\_noPass();  
 cout << "No-Passing Parameter Volume: " << volume << " units" << endl;  
 return 0;  
}

Passing Parameter Volume: 150 units

Enter Length, Width, Height Values:

5

10

3

No-Passing Parameter Volume: 150 units

Process finished with exit code 0

3. Write a C++ Program to show counter using Constructor.

#include <iostream>  
using namespace std;  
/\*  
show counter using Constructor.  
------------------------------------------------------  
 [x]create a class for room  
 [x]3 objects for calculations->(length, width, height)  
 [ ]AREA  
 [ ]VOLUME  
 [x]Use public access specifier and methods for calculations.  
\*/  
class counter{  
public:  
 int count, count\_incr{1};  
 int get\_counterIncr(){  
 int incr;  
 cout << "Enter counter increment: [ ]\b\b";  
 cin >> incr;  
 count\_incr = incr;  
 }  
 void counter\_up(){  
 count += count\_incr;  
 }  
 void counter\_down(){  
 count -= count\_incr;  
 }  
 int disp\_counter(){  
 cout << count << endl;  
 }  
};  
  
int main() {  
 int usr\_choice;  
 counter c{};  
 c.get\_counterIncr();  
 cout << "Enter [1]Count Up\n[2]Count Down\n" << endl;  
 cin >> usr\_choice;  
  
 if(usr\_choice = 1){  
 c.counter\_up();  
 }  
 else if(usr\_choice = 1){  
 c.counter\_down();  
 }  
 else{  
 cout << "Error: Invalid Input" << endl;  
 }  
 c.disp\_counter();  
return 0;  
}

Enter counter increment: [ ]

7

Enter counter increment: [Enter [1]Count Up

[2]Count Down

1

7

Process finished with exit code 0

4. Use Stack operations to

1. Display the elements in stack
2. get the size of stack
3. Print top of the stack
4. Pop some stack values and display the stack.