National University of Computer and Emerging Sciences



Task given in Class

22-04-2020

Object Oriented Programming

|  |  |
| --- | --- |
| Name | Muhammad Zain |
| Roll No. | 19F-0228 |
| INSTRUCTOR | Dr Danish |
| Semester | Spring 2020 |

Task 01:

# Source Code:

#include <iostream>

using namespace std;

class Test

{

private:

~Test() {}

};

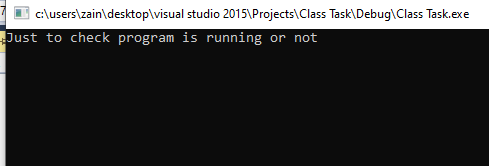
int main() {

cout << "Just to check program is running or not" << endl;

system("pause>0");

}

# Snip:



Task 02:

# Source Code:

#include <iostream>

using namespace std;

class Test

{

private:

~Test() {}

};

int main() {

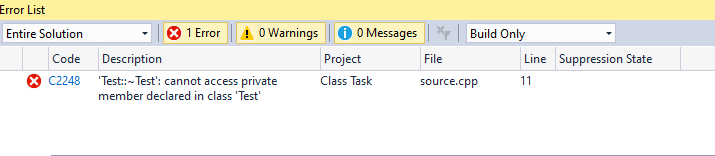
Test obj;

cout << "Just to check program is running or not" << endl;

system("pause>0");

}

# Snip:



Task 03:

# Source Code:

#include <iostream>

using namespace std;

class Test

{

private:

~Test() {}

};

int main() {

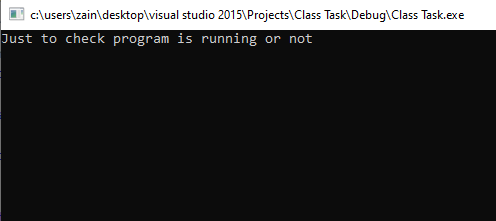
Test \*Pointer\_obj;

cout << "Just to check program is running or not" << endl;

system("pause>0");

}

# Snip:



Task 04:

# Source Code:

#include <iostream>

using namespace std;

class Test

{

private:

~Test() {}

};

int main() {

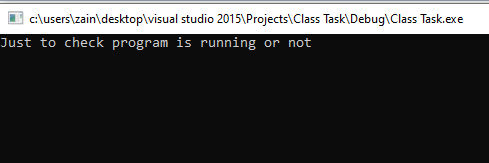
Test\* t = new Test;

cout << "Just to check program is running or not" << endl;

system("Pause>0");

}

# Snip:



Task 05:

# Source Code:

#include <iostream>

using namespace std;

class Test

{

private:

~Test() {}

};

int main() {

Test\* pointerobj = new Test;

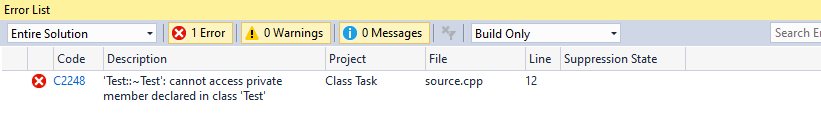
delete pointerobj;

cout << "Just to check program is running or not" << endl;

system("Pause>0");

}

# Snip:



Task 06:

# Source Code:

#include<iostream>

using namespace std;

class Test

{

private:

~Test() {}

friend void d\_Test(Test\*);

};

void d\_Test(Test\* pointer) {

delete pointer;

}

int main() {

Test\* pointerobj = new Test;

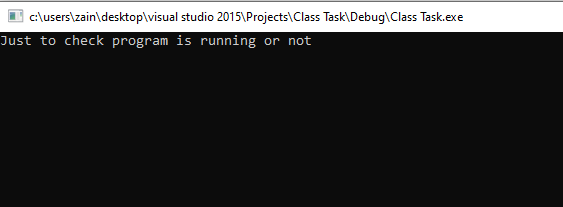
cout << "Just to check program is running or not" << endl;

d\_Test(pointerobj);

system("Pause>0");

}

# Snip:



Task 07:

# Source Code:

#include<iostream>

using namespace std;

class Sum\_Operator

{

private:

int real, imaginary;

public:

Sum\_Operator operator ++()

{

real++;

imaginary++;

return \*this;

}

Sum\_Operator()

{

cout << "Input the real number "<<endl;

cin >> real;

cout << "Input the imaginary part "<<endl;

cin >> imaginary;

}

void display()

{

cout << "After overloading operator = " << real << " + " << imaginary << " i ";

}

};

int main()

{

Sum\_Operator obj\_S;

++obj\_S;

obj\_S.display();

system("pause>0");

}

# Snip:

