**Assignment 11**

1. Write a C++ program to create an array of objects and get/put data in all array elements

Code:

#include<iostream>

using namespace std;

class Pilot

{

int age,id;

char name[30];

public:

void getdata();

void putdata();

};

void Pilot::getdata(){

cout<<"Enter Id : ";

cin>>id;

cout<<"Enter Pilot Name : ";

cin>>name;

cout<<"Enter Pilot Age : ";

cin>>age;

}

void Pilot::putdata(){

cout<<id<<" ";

cout<<name<<" ";

cout<<endl;

}

int main(){

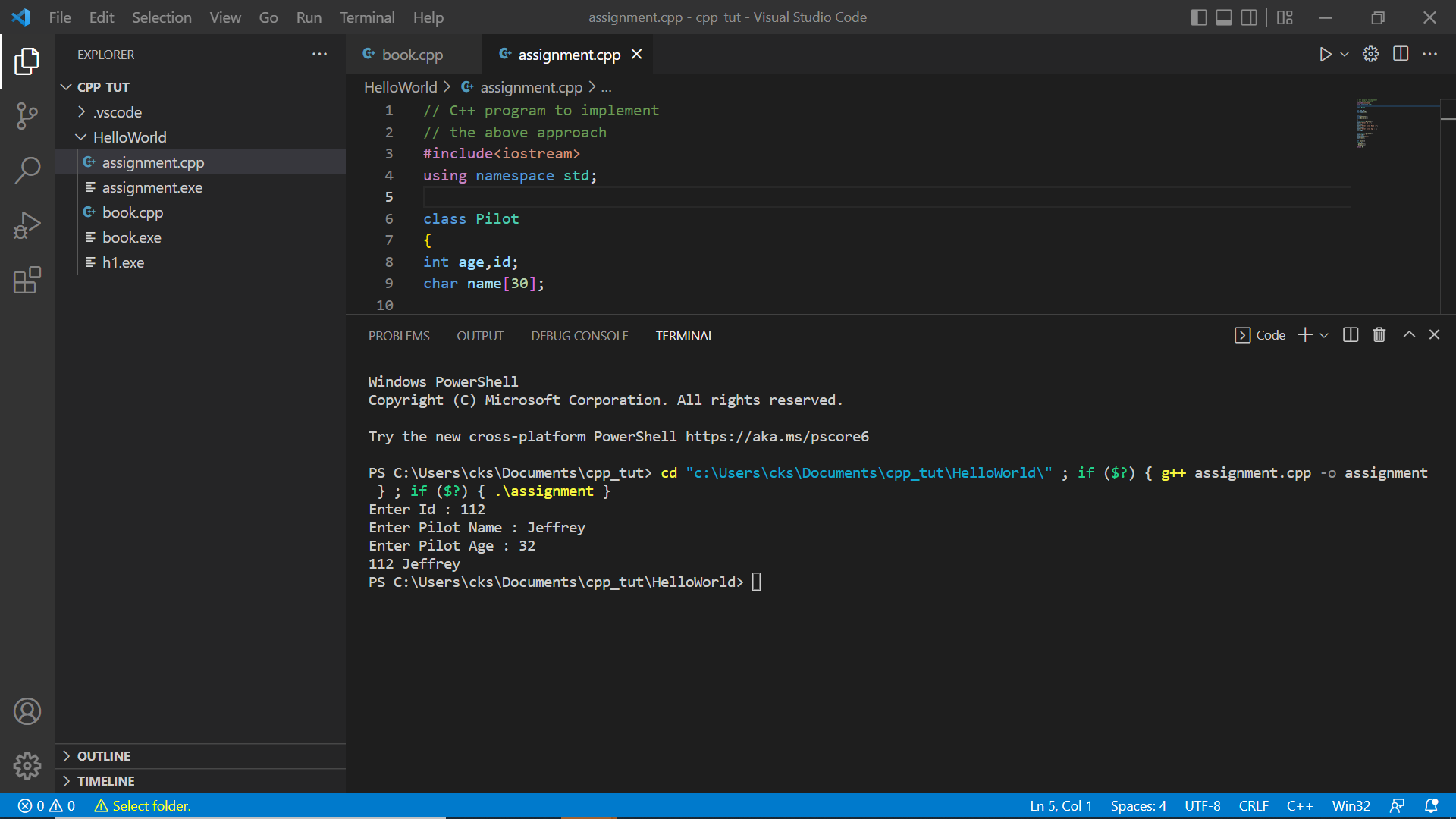
Pilot p;

p.getdata();

p.putdata();

return 0;

}

Output: 

1. Write a C++ program representing dynamic initialization of object using new operator.

Code:

#include <iostream>

using namespace std;

class C {

    int\* ptr;

public:

    C()

    {

        ptr = new int;

        \*ptr = 10;

    }

    void display()

    {

        cout << \*ptr << endl;

    }

};

int main()

{

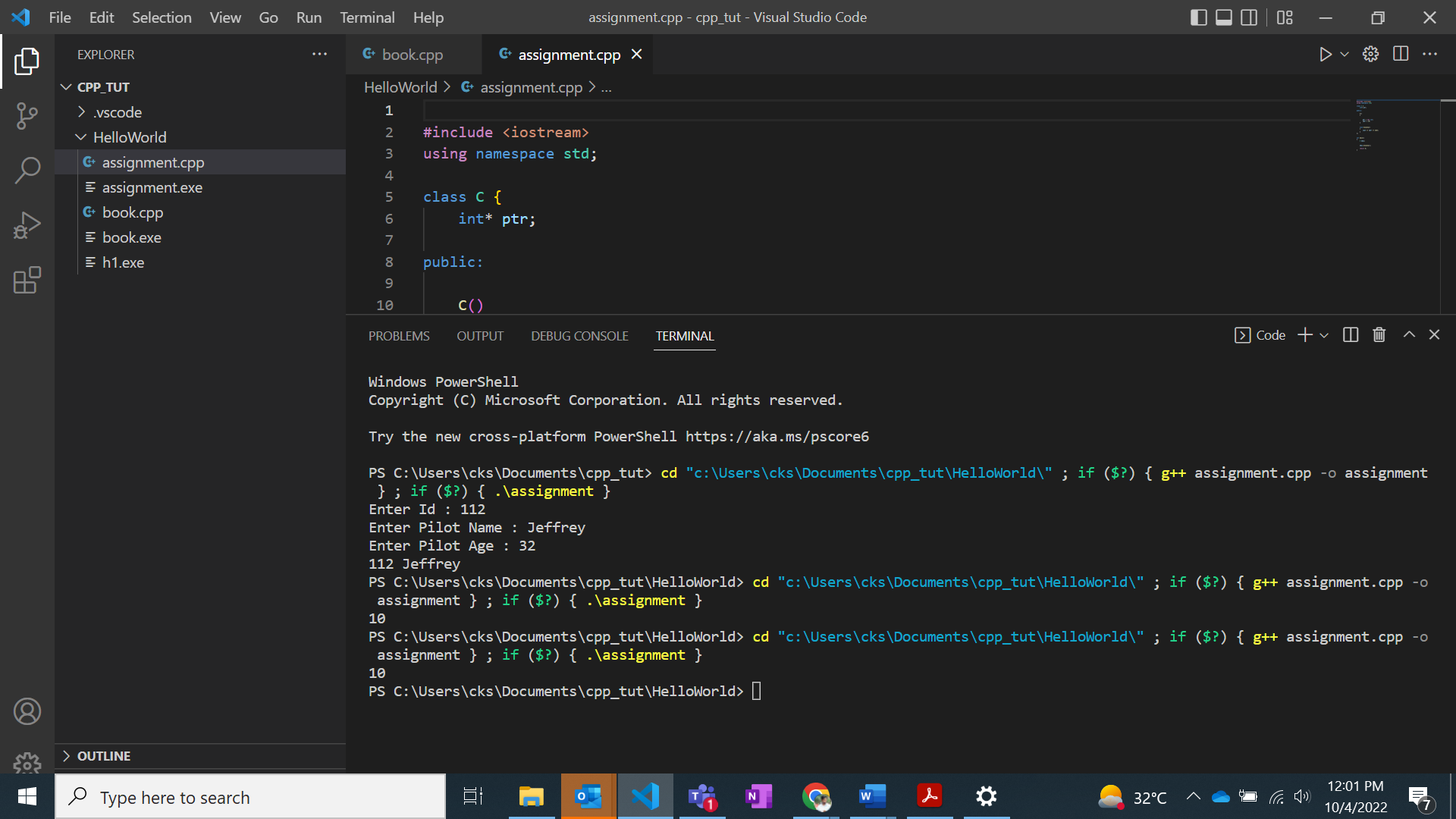
    C obj1;

    obj1.display();

    return 0;

}

Output:



1. Write a C++ program to check potential error for returning objects.

Code:

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

class sam {

   char \*s;

public:

   sam() { s = 0; }

   ~sam() { if(s) delete [] s; cout << "Freeing s........\n"; }

   void show() { cout << s << "\n"; }

   void set(char \*str);

};

void sam::set(char \*str)

{

  s = new char[strlen(str)+1];

  strcpy(s, str);

}

sam input()

{

  char instr[80] = "Testing..........";

  sam str;

  str.set(instr);

  return str;

}

int main()

{

  sam ob;

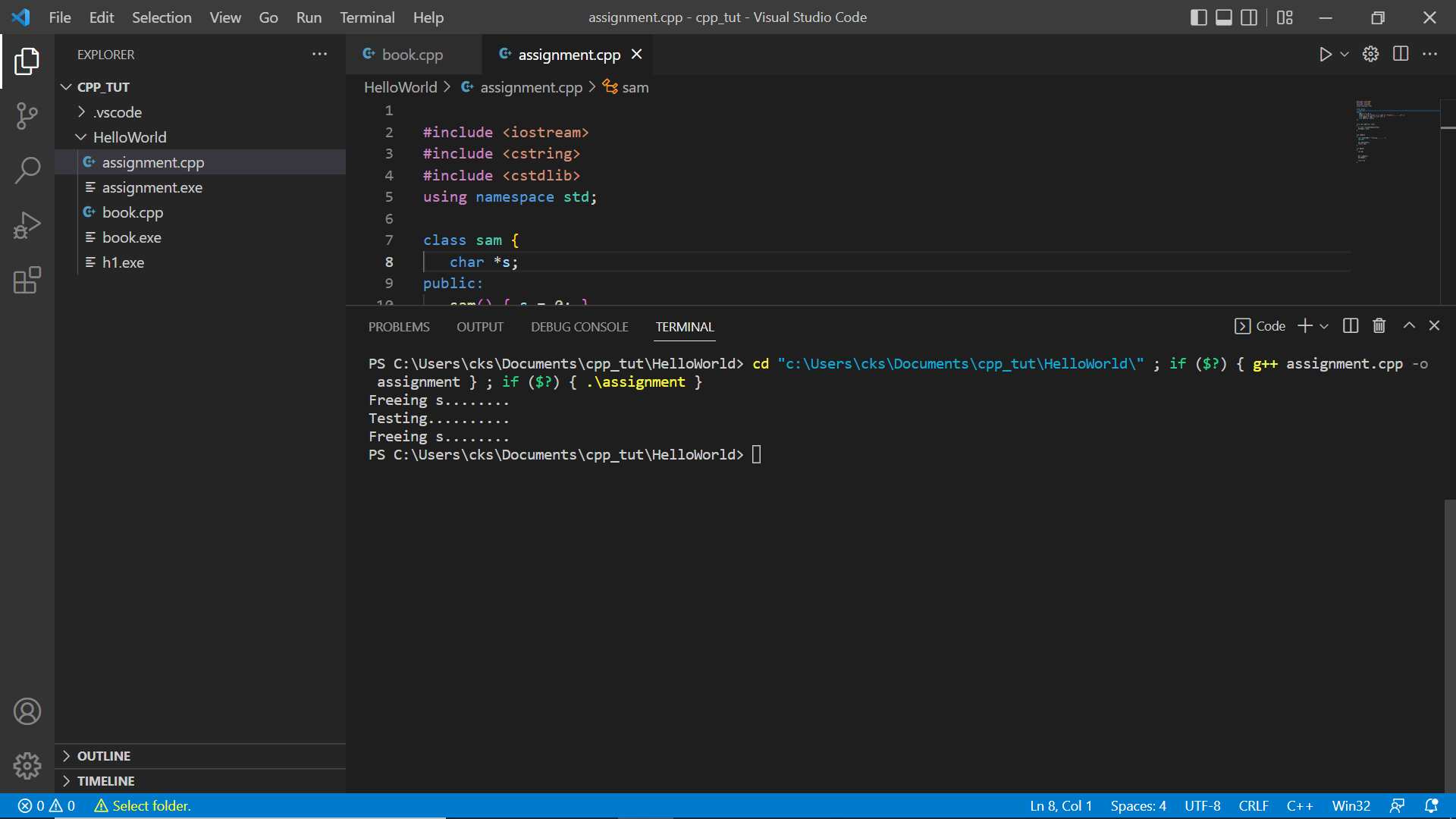
  ob = input();

  ob.show();

  return 0;

}

Output:



1. Write a C++ program demonstrates about the this \* pointer

**Code:**

#include<iostream>

using namespace std;

class Test

{

private:

int x;

int y;

public:

Test(int x = 0, int y = 0) { this->x = x; this->y = y; }

void setX(int a) { x = a; }

void setY(int b) { y = b; }

void destroy() { delete this; }

void print() { cout << "x = " << x << " y = " << y << endl; }

};

int main()

{

Test obj;

obj.destroy();

obj.print();

return 0;

}

**Output:**

