**9921103163**

**Nitin Chaudhary**

**F8**

**Tutorial 4**

**Ans -1**

#include <iostream>

using namespace std;

class TravelPlan

{

private:

    long PlanCode;

    string Place;

    int Number\_of\_travellers;

    int Number\_of\_buses;

    public :

    TravelPlan()

    {

        PlanCode = 1001;

        Place = "Agra";

        Number\_of\_travellers = 5;

        Number\_of\_buses = 1;

    }

    void NewPlan()

    {

        long PlanCode;

        string Place;

        int Number\_of\_travellers;

        int Number\_of\_buses;

        cout << "Enter PlanCode, Place and Number\_of\_travellers";

        cin >> PlanCode >> Place >> Number\_of\_travellers;

        if (Number\_of\_travellers < 20)

            Number\_of\_buses = 1;

        else if(Number\_of\_travellers >= 20 && Number\_of\_travellers < 40)

            Number\_of\_buses = 2;

        else

            Number\_of\_buses = 3;

    }

    void ShowPlan()

    {

        cout << "PlanCode=" << PlanCode << "\n Place=" << Place << "\nNumber\_of\_travellers=" << Number\_of\_travellers << "\nNumber\_of\_buses=" << Number\_of\_buses;

    }

};

int main()

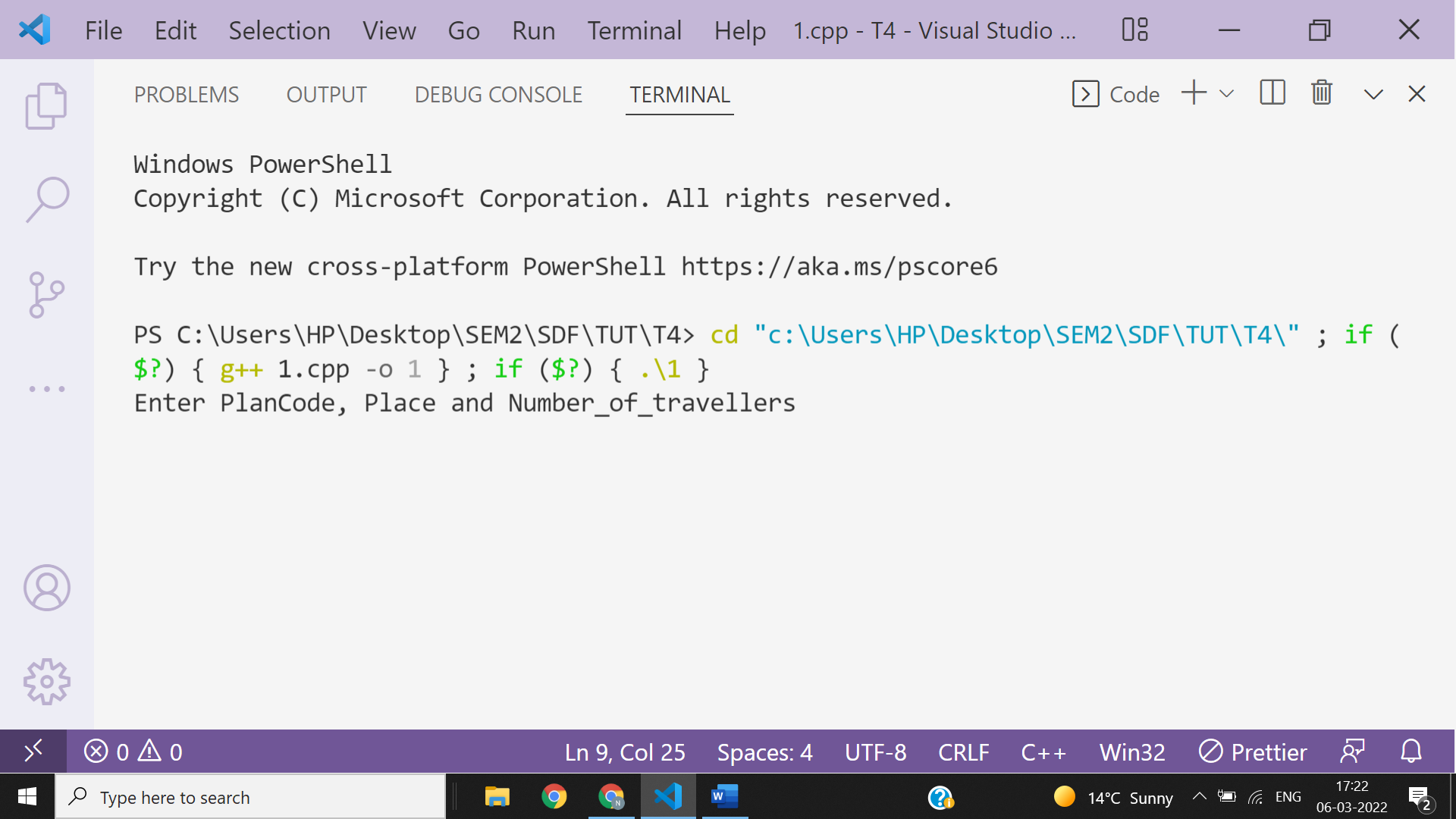
{

    TravelPlan obj1,obj2;

    obj1.NewPlan();

    obj1.ShowPlan();

}



**Q2. Explain the role of a default constructor? When is it considered equivalent to a parameterized constructor? Support your answer with examples.**

**Ans 2-**

**A default constructor is the one that takes no argument. It is automatically invoked when an object is created without providing any initial values . In case , the programmer has not defined a default constructor , the compiler automatically generates it.**

**For example.**

#include<iostream>

using namespace **std**;

class **A**

{

int a;

};

int **main**()

{

**A** obj;

return 0;

}

**A parameterized constructor with default argument is equivalent to a default constructor.**

**For example.**

#include<iostream>

using namespace **std**;

class **A**

{

    int i;

    int j;

    public:

**A**(int a=0,float b=1000.0);

    int a;

};

**A**::**A**(int a,float b)

{

    i=a;

    j=b;

}

int **main**()

{

**A** **o1**(23,23.50);

o1;

**A** o2;

return 0;

}

**Q3. What is a parameterized constructor? How is it useful?**

**Ans 3-** **A constructor that accepts parameters for its invocation is known as parameterized constructor.This helps you to assign initial value to an object at the time of its creation.**

**For example-**

#include<iostream>

using namespace **std**;

class **Test**{

    int ant;

    public:

**Test**(int i){

        ant=i;

    }

};

int **main**()

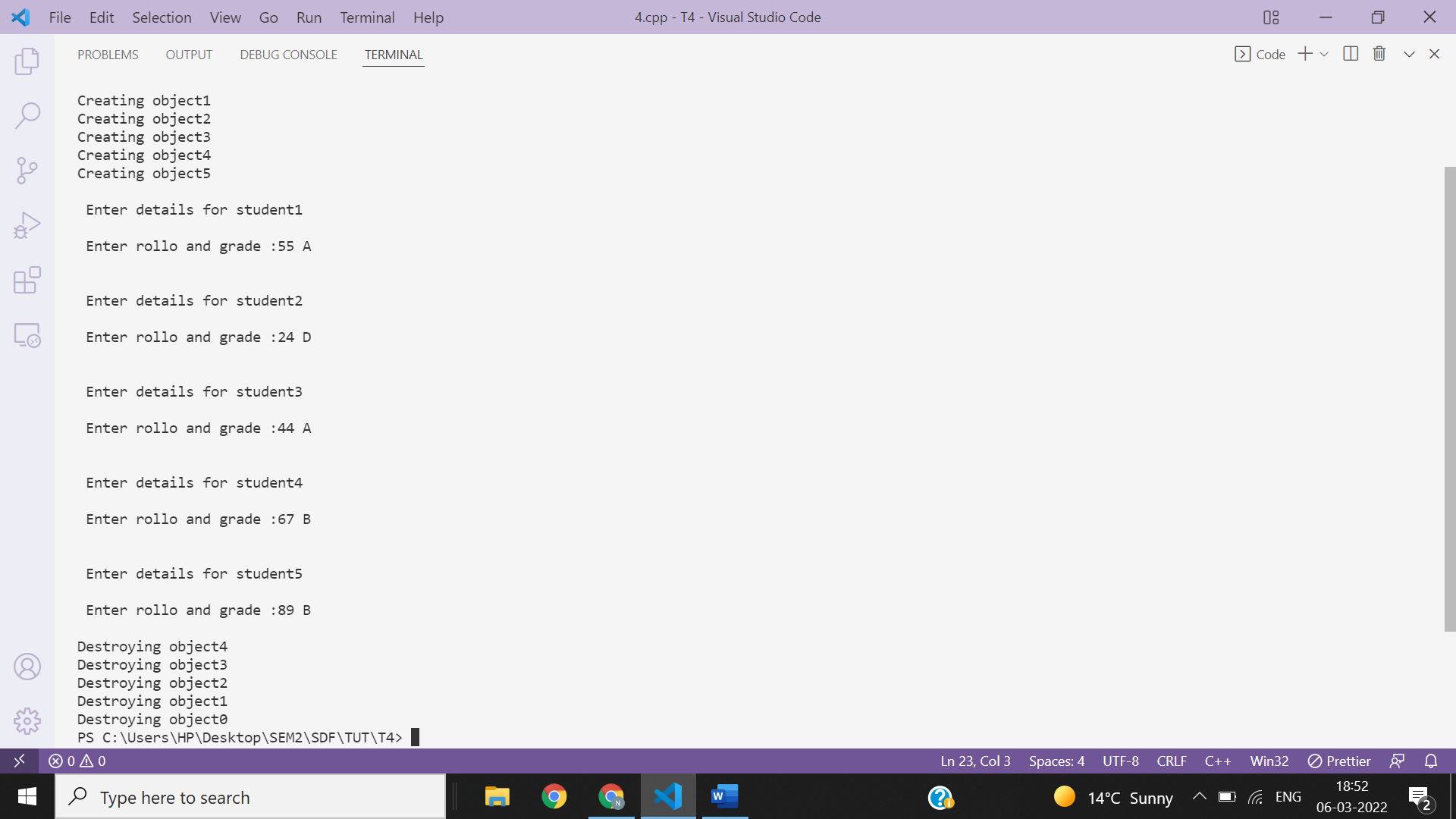
{

**Test** **ob1**(45);

}

**Ans 4**

**Output is-**



**Explanation-**

**First of all the programme executes default constructor as soon as object is created .**

**After that it executes destructor in the reverse order in which the constructor were called .**