**TUT-4**

**NAME-SANCHIT JAIN**

**BATCH-B-7**

**ENROLL-21103192**

**1.** #include <iostream>

#include <string.h>

using namespace std;

class Travel

{

private:

    long Code;

    char place[20];

    int trav, bus;

public:

    Travel() //default constructor

    {

        cout << "Default values invoked" << endl;

        Code = 1001;

        strcpy(place, "Agra");

        trav = 5;

        bus = 1;

    }

    void newPlan();

    void showPlan();

};

void Travel::newPlan()

{

    cout << "Enter New PlanCode: ";

    cin >> Code;

    cout << "Enter Place: ";

    cin.ignore(1, '\n');

    cin.getline(place, 20);

    cout << "Enter No. of Travellers: ";

    cin >> trav;

    if (trav < 20)

        bus = 1;

    else if (trav >= 20 && trav < 40)

        bus = 2;

    else if (trav >= 40)

        bus = 3;

}

void Travel::showPlan()

{

    cout << "PlanCode: " << Code << "\n"<< "Place: "<<place<<"\n "<<" NO.Travellers : "<<trav<<"\n "<<" No.Of buses : "<<bus<<endl;

}

int main()

{

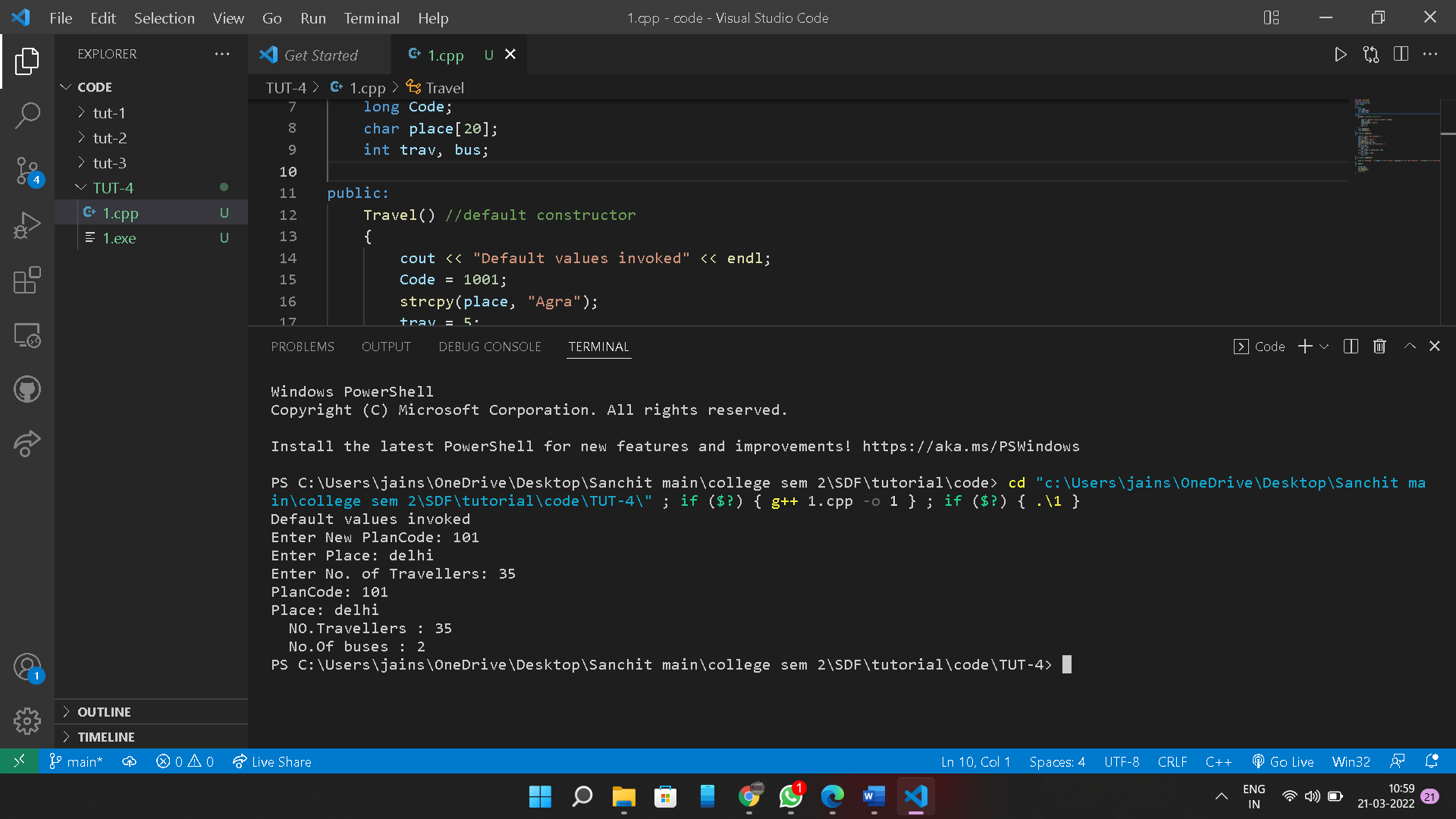
    Travel tb;

    tb.newPlan();

    tb.showPlan();

    return 0;

}



**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. A parameterised constructor with default arguments is the same as default con

For example, class A { i

nt i; float j;

public:

A(int a = 0,float b = 1000.0); //constructor with default argument };

A::A(int a,float b) //constructor definition { i = a; j = b; }

int main()

{

A o1(23,27);

A o2(); //takes default arguments

}

**3.**

The parameterized constructors are the constructors having a specific number of arguments to be passed. The purpose of a parameterized constructor is to assign user-wanted specific values to the instance variables of different objects.

4.

Output.

Creating object1

Creating object2

Creating object3

Creating object4

Creating object5

Enter details for student1

Enter rollo and grade :2

B

Enter details for student2

Enter rollo and grade :3

A

Enter details for student3

Enter rollo and grade :10

D

Enter details for student4

Enter rollo and grade :11

D

Enter details for student5

Enter rollo and grade :12

C

Destroying object4

Destroying object3

Destroying object2

Destroying object1

Destroying object0

**REASON: As we created an object array of size 5, objects were initialized one by one , thus, invoking the constructors 1 to 5. After exiting the programs destructors were invoked automatically, in the reverse order as per stack**