**TUTORIAL-4**

A1.

#include <iostream>

#include<string.h>

using namespace std;

class Traveller

{

private:

long Plancode;

char place[10];

int nof\_trav;

int nof\_b;

public:

Traveller()

{

Plancode=1001;

char p[]="Agra";

strcpy(place,p);

nof\_trav=5;

nof\_b=1;

}

void NewPlan()

{

cout<<"Enter Plan Code:";

cin>>Plancode;

getchar();

cout<<"Enter Place:";

gets(place);

cout<<"Enter Number of Travellers:";

cin>>nof\_trav;

if(nof\_trav<20)

nof\_b=1;

else if(nof\_trav>=20&&nof\_trav<40)

nof\_b=2;

else

nof\_b=3;

}

void ShowPlan()

{

cout<<"Plancode:"<<Plancode<<endl;

cout<<"Place:";

puts(place);

cout<<"Number of Travellers:"<<nof\_trav<<endl;

cout<<"Number of Buses:"<<nof\_b;

}

};

int main()

{

Traveller obj;

cout<<"Eneter details"<<endl;

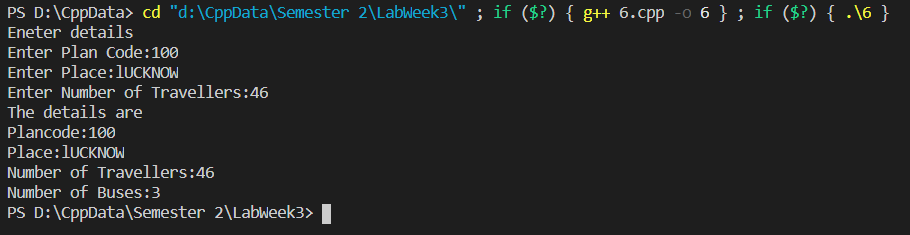
obj.NewPlan();

cout<<"The details are"<<endl;

obj.ShowPlan();

return 0;

}

OUTPUT:

A2:

The default constructor is used to initialize the member variables when an object is created. It does not takes any argument. When arguments are passed to the constructor then it is called parameterized constructor.

#include <iostream>

#include<string.h>

using namespace std;

class marks

{

int mark;

float grade;

public:

marks()//default constructor

{

mark=0;

grade=7.5;

}

marks(int a,float b)//parameterized constructor

{

mark=a;

grade=b;

}

};

int main()

{

marks obj1;

marks obj2(90,8.5);

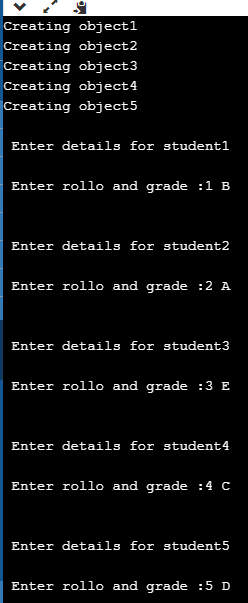
return 0;

}

A3.

A constructor which takes arguments is called a parameterized constructor. It can be helpful in assigning values to the data variables by passing values in the form of arguments.

A4.

The data members get initialised in constructors by default values and then values are inputted using a function. Finally, destructor is called.

