Exercise1

Define a **struc**t Student with the following specification

id	integer
name	string
char	grade

with two functions:

Student get_*data(*Student*)* Function to accept values for id, name and grade and return an object.

Void *Show_data(*Student) Function to display all the data members on the screen.

Sample Run:

```
Enter ID: 1439
Enter name: Mohamed
Enter Grade: A
ID:1439
Name: Mohamed
Grade: A
```

Answer:

```
#include <iostream>

#include <string>

using namespace std;

struct Student

{

int id;

string name;

char grade;

};

Student  get_data (Student  in_student);

void show_data(Student);
```

```
int main ()

{

Student  Student1;

Student1 = get_data (Student1);

show_data(Student1);

return 0;

}

Student get_data (Student in_student)

{

cout<<"Enter ID: "; cin>> in_student.id;

cout<<"Enter name: "; cin>> in_student.name;

cout<<"Enter Grade: "; cin>> in_student.grade;

return (in_student);

}

void show_data(Student  Student1)

{

cout<< "ID:"<<Student1.id<<endl;

cout<<"Name: "<<Student1.name<< endl;

cout<<"Grade: "<<Student1.grade;

   }
```

Exercise2:

Define a **struct** Time with the following specification

hour	integer
minute	integer
second	integer

with two functions:

void printuniversal*(Time)* Function to accept values for hour, minute and second.

Void printStandard*(Time*) Function to display all the data members on the screen with Standard time.

Sample Run:

```
Dinner will be held at 18:30:00 universal time,
which is 6:30:00 PM standard time.
```

Answer:

```
#include <iostream>

#include <iomanip>

using namespace std;

struct Time

{

int hour; // 0-23 (24-hour clock format)

int minute; // 0-59

int second; // 0-59

};

void printUniversal( const Time & ); // prototype

void printStandard( const Time & );
```

```
int main()
{
Time dinnerTime; // variable of new type Time
dinnerTime.hour = 18; // set hour member of dinnerTime
dinnerTime.minute = 30; // set minute member of dinnerTime
dinnerTime.second = 0; // set second member of dinnerTime
cout << "Dinner will be held at ";
printUniversal( dinnerTime );
cout << " universal time,\nwhich is ";
printStandard( dinnerTime );
cout << " standard time.";
return 0;
}
void printUniversal( const Time &t ) // print time in universal-time format
 {
cout << setfill( '0' ) << setw( 2 ) << t.hour << ":"<< setw( 2 ) << t.minute << ":"<< setw( 2 ) << t.second;
}
void printStandard( const Time &t ) // print time in standard-time format
{
cout <<(( t.hour == 0 | | t.hour == 12 ) ?  12 : t.hour % 12 );
cout<< ":" << setfill( '0' ) << setw( 2 ) << t.minute << ":" << setw( 2 ) << t.second ;
cout<< ( t.hour < 12 ? " AM" : " PM" );
   }
```