Exercise1

```
// classes first example
#include <iostream>
using namespace std;
class Rectangle
 {
   int width, height;
  public:
   void set_values (int,int);
   int area()
            {
            return width*height;
            }
};
void Rectangle::set_values (int x, int y)
 {
  width = x;
  height = y;
}
int main ()
{
  Rectangle rect;
  rect.set_values (3,4);
```

```
  cout << "area: " << rect.area();
  return 0;
}
```

Exercise2:

```
/* C++ program to create a simple class and object.*/
#include  <iostream>
using namespace std;
class Hello
{
   public:
      void sayHello()
      {
            cout << "Hello World" << endl;
      }
};
int main()
{
   Hello h;
   h.sayHello();
   return 0;
}
```

Exercise3:

```
/ /member initialization
#include <iostream>
using namespace std;
```

```
class Circle
 {
    double radius;
  public:
    Circle(double r) : radius(r) { }
    double area()
              {
              return radius*radius*3.14;
              }
};
class Cylinder
{
    Circle base;
    double height;
  public:
    Cylinder(double r, double h) : base (r), height(h) {}
    double volume()
              {
              return base.area() * height;
              }
};
int main ()
{
  Cylinder  cylinder (10,20);
```

```
 cout << " volume of cylinder is: " << cylinder.volume() << '\n';
 return 0;
}
```

Exercise4:

```
//Using classes to add 2 numbers
#include <iostream>
using namespace std;
//class definition
class Numbers
{
	private:
		int a;
		int b;
	public:
		//member function declaration
		void readNumbers(void);
		void printNumbers(void);
		int calAddition(void);
};
//member function definitions
void Numbers::readNumbers(void)
{
	cout<<"Enter first number: ";
	cin>>a;
	cout<<"Enter second number: ";
```

```
        cin>>b;
}
void Numbers::printNumbers(void)
{
        cout<<"a= "<<a<<",b= "<<b<<endl;
}
int Numbers::calAddition(void)
{
        return (a+b);
}
int main() //main function
{
        Numbers num; //declaring object
        int add; //variable to store addition
        num.readNumbers(); //take input
        add=num.calAddition(); //find addition
        num.printNumbers(); //print numbers
        cout<<"Addition/sum= "<<add<<endl; //print addition
        return 0;
}
```

Exercise5:

```
/*C++ program to create class for a student.*/
#include <iostream>
using namespace std;
class student
```

```
{
        private:
                char  name[30];
                int   IDNo;
                int   total;
                public:
                //member function to get student's details
                void getDetails(void);
                //member function to print student's details
                void putDetails(void);
};
//member function definition, outside of the class
void student::getDetails(void)
{
        cout << "Enter name: " ;
        cin >> name;
        cout << "Enter ID number: ";
        cin >> IDNo;
        cout << "Enter total marks out of 500: ";
        cin >> total;
        }
//member function definition, outside of the class
void student::putDetails(void)
{
        cout << "Student details:\n";
```

```
        cout << "Name:"<< name << ", ID Number:" << IDNo << ",Total:" << total ;
}
int main()
{
        student std;                //object creation
        std.getDetails();
        std.putDetails();
        return 0;
}
```

Exercise6:

```
/*C++ program to create student class, read and print 10 student's details (Example of array of objects).*/
 #include <iostream>
using namespace std;
 #define MAX= 10
 class student
{
   private:
      char  name[30];
      int   IDNo;
      int   total;
      float perc;
   public:
      //member function to get student's details
      void getDetails(void);
      //member function to print student's details
```

```
        void putDetails(void);
};
 //member function definition, outside of the class
void student::getDetails(void){
    cout << "Enter name: " ;
    cin >> name;
    cout << "Enter ID number: ";
    cin >> IDNo;
    cout << "Enter total marks out of 500: ";
    cin >> total;
}
//member function definition, outside of the class
void student::putDetails(void){
    cout << "Student details:\n";
    cout << "Name:"<< name << ",ID Number:" << IDNo << ",Total:" << total ;
}
int main()
{
    student std[MAX];       //array of objects creation
    int n,loop;
     cout << "Enter total number of students: ";
    cin >> n;
         for(loop=0;loop< n; loop++){
        cout << "Enter details of student " << loop+1 << ":\n";
        std[loop].getDetails();
```

```
    }
        cout << endl;

    for(loop=0;loop< n; loop++)
{
        cout << "Details of student " << (loop+1) << ":\n";
        std[loop].putDetails();
    }
        return 0;
}
```

Exercise7:

```
/*C++ program to create class to read and add two times.*/
#include <iostream>
using namespace std;
 class Time
{
private:
    int hours;
    int minutes;
    int seconds;
 public:
    void getTime(void);
    void putTime(void);
    void addTime(Time T1,Time T2);
};
```

```
void Time::getTime(void)
{
   cout << "Enter time:" << endl;
   cout << "Hours? ";    cin>>hours;
   cout << "Minutes? ";  cin>>minutes;
   cout << "Seconds? ";  cin>>seconds;
}
 void Time::putTime(void)
{
   cout << endl;
   cout << "Time after add: ";
   cout << hours << ":" << minutes << ":" << seconds << endl;
}
 void Time::addTime(Time T1,Time T2)
{

   this->seconds=T1.seconds+T2.seconds;
   this->minutes=T1.minutes+T2.minutes + this->seconds/60;;
   this->hours= T1.hours+T2.hours + (this->minutes/60);
   this->minutes %=60;
   this->seconds %=60;
}
int main()
{
```

```
   Time T1,T2,T3;
   T1.getTime();
   T2.getTime();
   //add two times
   T3.addTime(T1,T2);
   T3.putTime();
   return 0;
}
```

Exercise8:

```
/*C++ program to demonstrate example of Default Constructor or No argument.*/
#include <iostream>
using namespace std;
 //Class declaration.
Class Point
{
   //Private block  to declare data member( X,Y ) of integer type.
   private:
      int X;
      int Y;
    //Public block of member function to access data members.
   public:
      //Declaration of default or no argument constructor to initialize data members.
         Point ();
      void    Input();       //To take input from user.
      void    Display(); //To display output on screen.
```

```
    };   //End of class
 //Definition of constructor.
Point::Point()
{
   X = 0;
   Y = 0;
}
//Definition of Input() member function.
void Point:: Input()
{
   cout << "Enter Value of X: "; cin >> X;
   cout << "Enter Value of Y: "; cin >> Y;
}
 //Definition of Display() member function.
void Point:: Display()
{
   cout << endl << "X: " << X;
   cout << endl << "Y: " << Y << endl;
}
 int main()
{
  Point d ; //Constructor automatically call when object is created.
   //Display value of data member.
   cout << endl <<"Method 1: " << endl;
   cout << "Value after initialization : " ;
```

```
    d.Display();
     d.Input();
    cout << "Value after User Input : ";
    d.Display();
     //We can also create object like this
    Point d1 = Point();
        //Display value of data member.
    cout << endl << "Method 2: " << endl;
    cout << "Value after initialization : ";
    d1.Display();
       return 0;
}
```