Classes examples

1. Class and Object example
   ==========

```
#include <iostream>

using namespace std;

class Student {

  public:

    int id;//data member (also instance variable)

    string name;//data member(also instance variable)

};

int main() {

   Student s1; //creating an object of Student

   s1.id = 201;

   s1.name = "Sonoo Jaiswal";

   cout<<s1.id<<endl;

   cout<<s1.name<<endl;

   return 0;

}
```

Initialize data through methods

```
#include <iostream>

using namespace std;

class Student {

  public:

    int id;//data member (also instance variable)

    string name;//data member(also instance variable)
```

```
    void insert(int i, string n)
    {
        id = i;
        name = n;
    }
    void display()
    {
        cout<<id<<"  "<<name<<endl;
    }
};
int main(void) {
   Student s1; //creating an object of Student
   Student s2; //creating an object of Student
   s1.insert(201, "Sonoo");
   s2.insert(202, "Nakul");
   s1.display();
   s2.display();
   return 0;
}
```

Store and Display Employee Information

```
#include <iostream>
using namespace std;
class Employee {
  public:
```

```
    int id;//data member (also instance variable)
    string name;//data member(also instance variable)
    float salary;
    void insert(int i, string n, float s)
     {
        id = i;
        name = n;
        salary = s;
     }
    void display()
     {
        cout<<id<<"  "<<name<<"  "<<salary<<endl;
     }
};
int main(void) {
   Employee e1; //creating an object of Employee
   Employee e2; //creating an object of Employee
   e1.insert(201, "Sonoo",990000);
   e2.insert(202, "Nakul", 29000);
   e1.display();
   e2.display();
   return 0;
}
```

Constructor

Default constructor

```
#include <iostream>
using namespace std;
class Employee
 {
   public:
       Employee()
       {
          cout<<"Default Constructor Invoked"<<endl;
       }
};
int main(void)
{
    Employee e1; //creating an object of Employee
    Employee e2;
    return 0;
}
```

Parameterised Constructor

```
#include <iostream>
using namespace std;
class Employee {
   public:
      int id;//data member (also instance variable)
      string name;//data member(also instance variable)
      float salary;
```

```
    Employee(int i, string n, float s)
    {
        id = i;
        name = n;
        salary = s;
    }
    void display()
    {
        cout<<id<<"  "<<name<<"  "<<salary<<endl;
    }
};
int main(void) {
   Employee e1 =Employee(101, "Sonoo", 890000); //creating an object of Employee
   Employee e2=Employee(102, "Nakul", 59000);
   e1.display();
   e2.display();
   return 0;
}
```

Calculate the area of a rectangle and display it

```
#include <iostream>
using namespace std;

class Area
```

```
{
   private:
      int length;
      int breadth;

   public:
      // Constructor
      Area(): length(5), breadth(2){ }

      void GetLength()
      {
         cout << "Enter length and breadth respectively: ";
         cin >> length >> breadth;
      }

      int AreaCalculation() {  return (length * breadth);  }

      void DisplayArea(int temp)
      {
         cout << "Area: " << temp;
      }
};

int main()
{
```

```
    Area A1, A2;
    int temp;

    A1.GetLength();
    temp = A1.AreaCalculation();
    A1.DisplayArea(temp);

    cout << endl << "Default Area when value is not taken from user" << endl;

    temp = A2.AreaCalculation();
    A2.DisplayArea(temp);

    return 0;
}

// public member variable accessible from anywhere outside the class

#include <iostream>

using namespace std;

class Line {
  public:
     double length;
     void setLength( double len );
```

```
    double getLength( void );
};

// Member functions definitions
double Line::getLength(void) {
  return length ;
}

void Line::setLength( double len ) {
  length = len;
}

// Main function for the program
int main( ) {
  Line line;

  // set line length
  line.setLength(6.0);
  cout << "Length of line : " << line.getLength() <<endl;

  // set line length without member function
  line.length = 10.0; // OK: because length is public
  cout << "Length of line : " << line.length <<endl;
  return 0;
}
```

```
//private member variable

#include <iostream>

using namespace std;

class Box {
  public:
    double length;
    void setWidth( double wid );
    double getWidth( void );

  private:
    double width;
};

// Member functions definitions
double Box::getWidth(void) {
  return width ;
}

void Box::setWidth( double wid ) {
  width = wid;
}
```

```
// Main function for the program
int main( ) {
   Box box;

   // set box length without member function
   box.length = 10.0; // OK: because length is public
   cout << "Length of box : " << box.length <<endl;

   // set box width without member function
   // box.width = 10.0; // Error: because width is private
   box.setWidth(10.0);  // Use member function to set it.
   cout << "Width of box : " << box.getWidth() <<endl;

   return 0;
}
#include <iostream>
#include <iostream>
using namespace std;

class Box {
   protected:
      double width;
};
```

```
class SmallBox:Box // SmallBox is the derived class. {
  public:
    void setSmallWidth( double wid );
    double getSmallWidth( void );
};

// Member functions of child class
double SmallBox::getSmallWidth(void) {
  return width ;
}

void SmallBox::setSmallWidth( double wid ) {
  width = wid;
}

// Main function for the program
int main( ) {
  SmallBox box;

  // set box width using member function
  box.setSmallWidth(5.0);
  cout << "Width of box : "<< box.getSmallWidth() << endl;

  return 0;
}
```