SLIIT

Discover Your Future

IT1050- Object Orientation

Lecture-05

Classes and Objects in C++

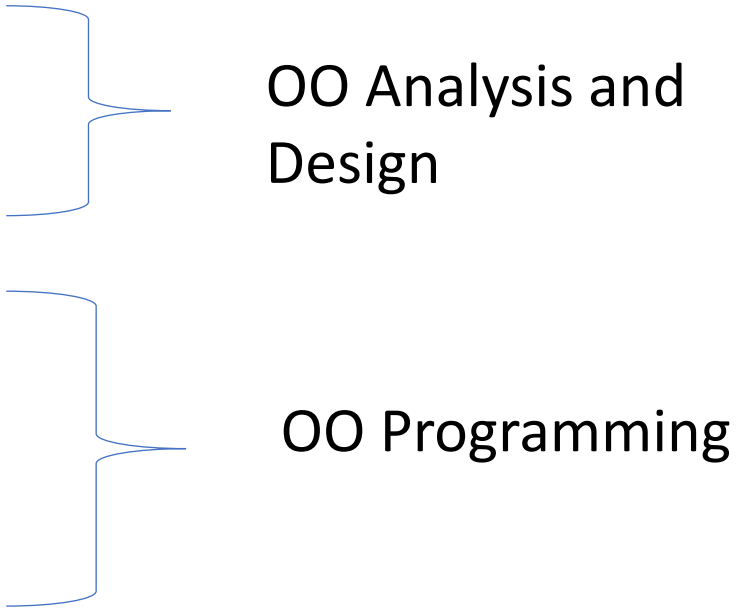
Learning Outcomes

At the end of the Lecture students should be able to;

- Implement a class and create objects using C++.

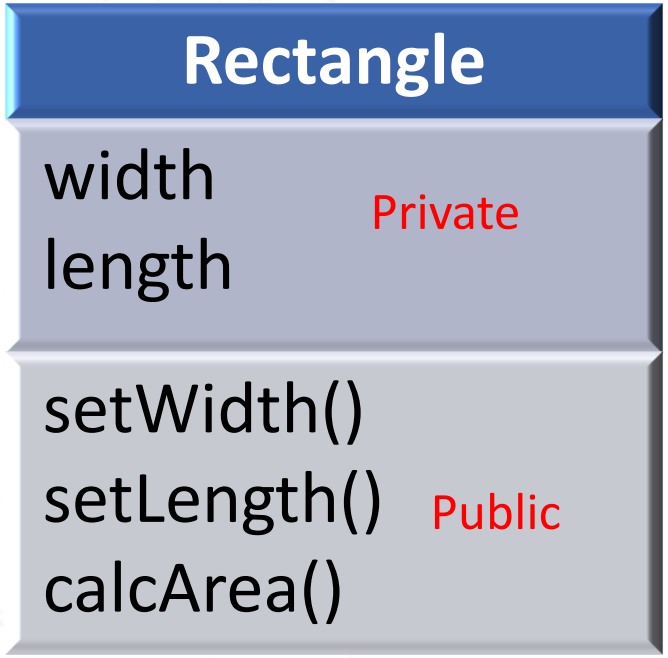
Recalling Steps in OOP

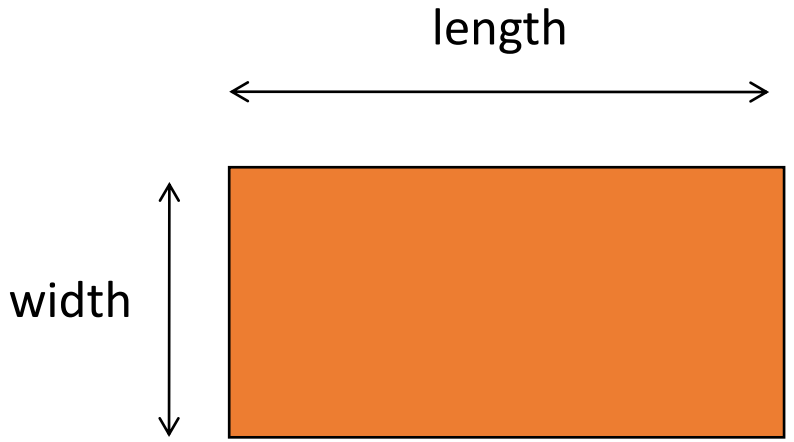
- Analyse the Problem
- Identify Objects
- Develop Classes
- Create Objects
- Build the Solution



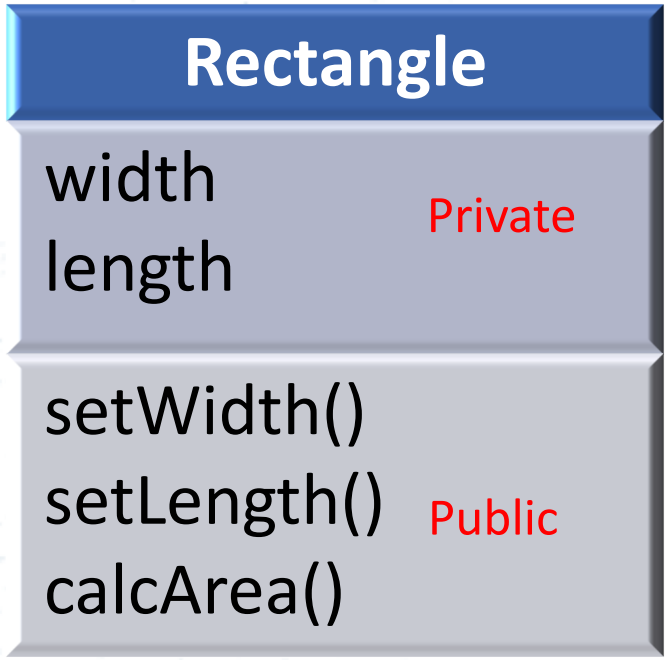
How to write an Object Oriented Program ?

Example-1 - Rectangle Class





Rectangle Class in C++

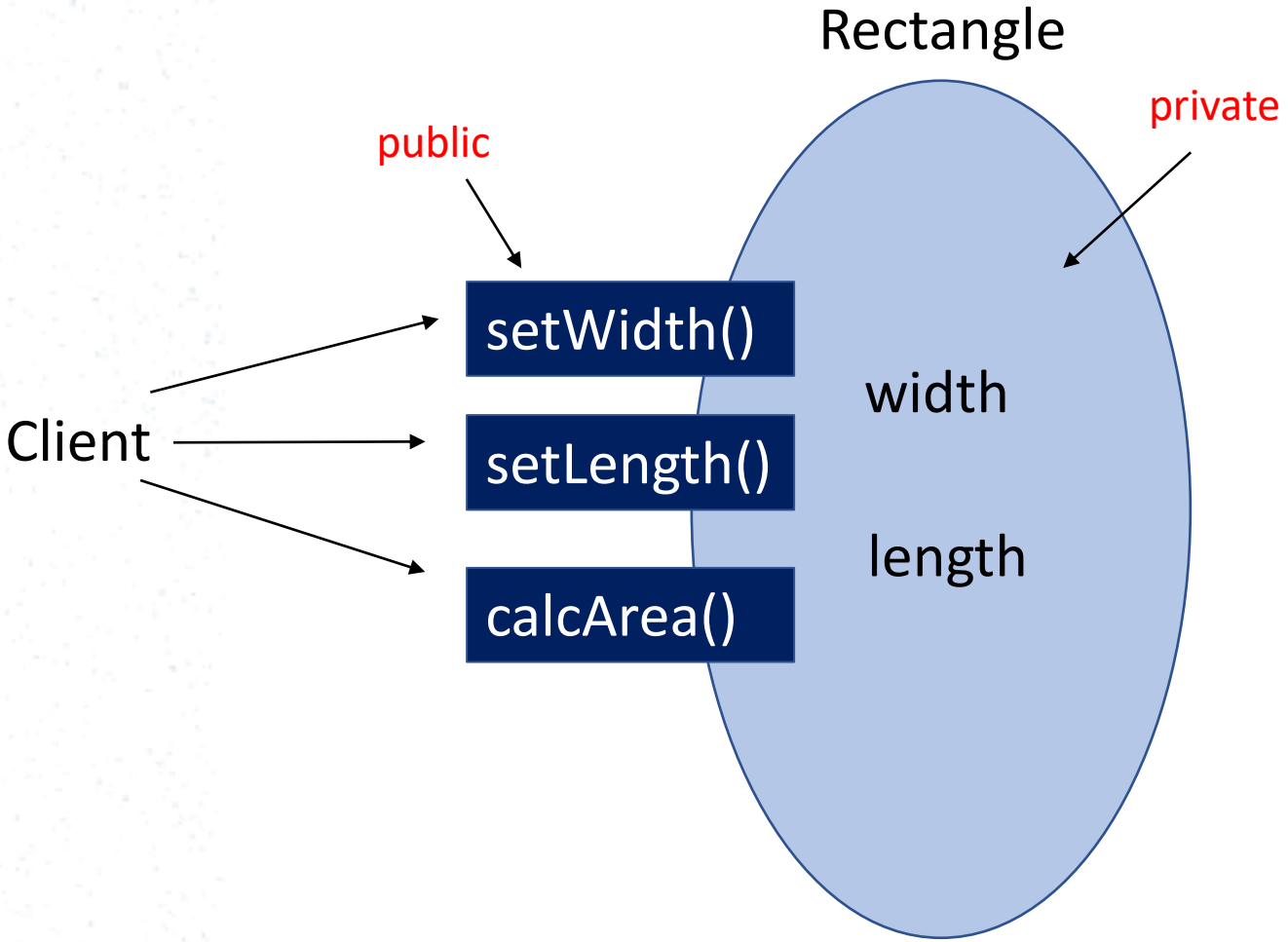



```
//C++ coding for Rectangle class
class Rectangle {
   private:
          int width;
           int length;
    public:
          void setWidth(int no);
           void setLength(int no);
           int calcArea();
};
```

Private & Public

- The private part of the definition specifies the properties (data members) of a class.
- These are hidden from outside the class and can only be accessed through the methods (operations/functions) defined for the class.
- The public part of the definition specifies the methods as function prototypes.
- These methods as they are called, can be accessed by the main program.

Private & Public

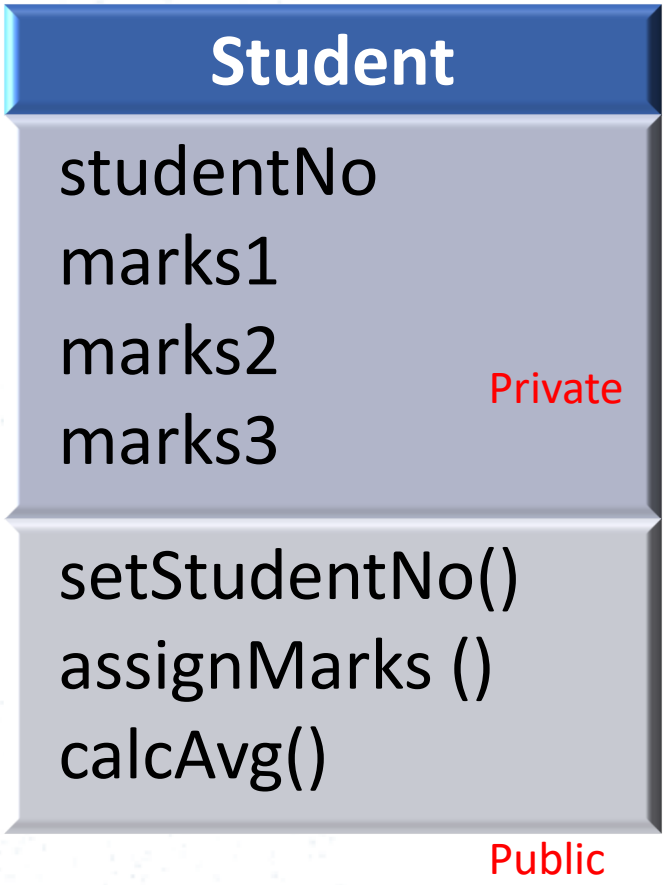


Rectangle Class in C++

```
//C++ coding for Rectangle
class Rectangle {
   private :
          int width;
          int length;
    public:
          void setWidth(int no);
          void setLength(int no);
          int calcArea();
};
```

```
void Rectangle::setWidth(int no) {
     width = no;
}

void Rectangle::setLength(int no) {
     length= no;
}
int Rectangle::calcArea() {
     int area = length * width;
     return area;
}
```

Student class – Activity 1



Implement the Student class in C++.

Answer : Student Class in C++

```
//C++ coding for Student
class Student {
   private :
      int StudentNo;
      int marks1;
      int marks2;
      int marks3;
    public:
      void setStudentNo(int no);
      void assignMarks(int n1, int n2, int n3);
      float calcAvg();
};
```

```
void Student::setStudentNo(int no) {
    width = no;
}

void Student :: assignMarks(int n1, int n2, int n3);{
    marks1 = n1;
    marks2 = n2;
    marks3 = n3;
}
float Student ::calcAvg() {
    float average = (marks1+marks2+marks3)/3.0;
    return average;
}
```

Creating Objects

```
Class_name   Object_name;
```

e.g: Rectangle rect1; // single object

Rectangle rect1, rect2; // multiple objects

Rectangle rectangles[5]; // array of objects

Note : Use C++ rules for identifiers when naming objects

Creating Objects

```
Rectangle rec1, rec2;
```

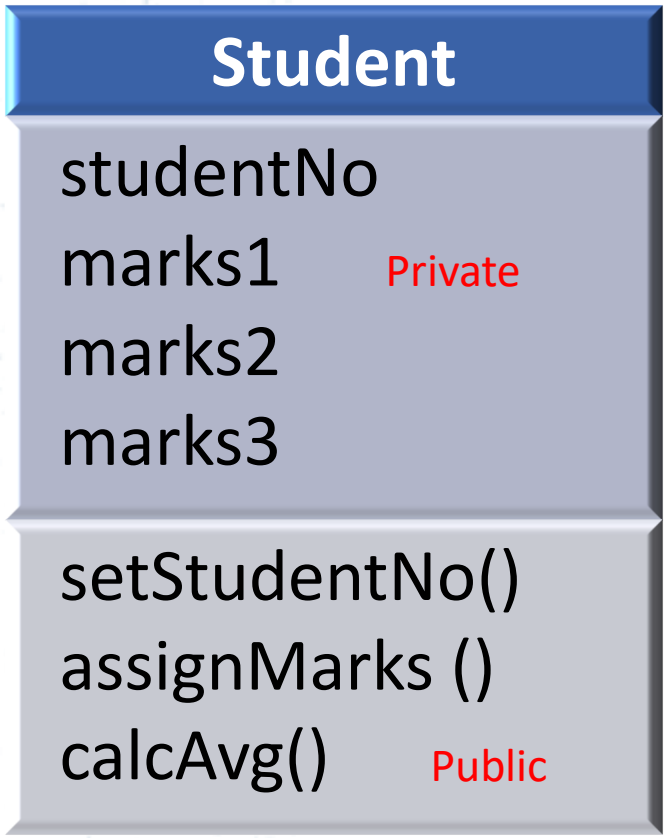
rec1 : Rectangle
width = 10 length = 20

rec2 : Rectangle
width = 5 length = 10

Accessing Public Methods

```
#include <iostream>
using namespace std;
int main() {
    Rectangle rec1, rec2;

    rec1.setWidth(10);
    rec1.setLength(20);

    rec2.setWidth(5);
    rec2.setLength(10);

    cout << rec1.calcArea() << endl;
    cout << rec2.calcArea() << endl;
    return 0;
}
```

Student class – Activity 2



Create two Student Objects.
Calculate and print their averages.

Student Objects

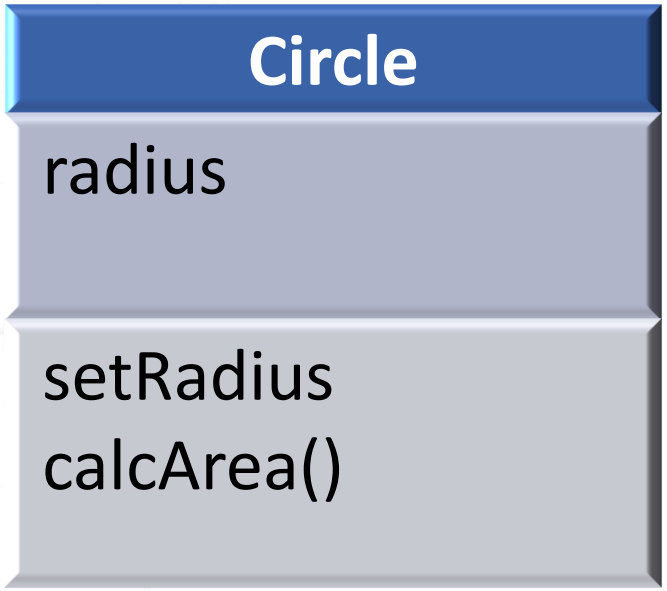
std1: Student
studentNo = 1023 marks1= 50 marks2= 60 marks3 = 70

std2: Student
studentNo= 2345 marks1= 70 marks2= 80 marks3 = 75

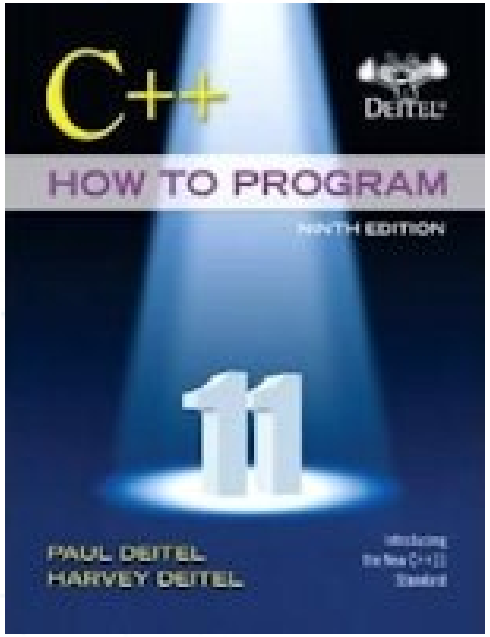
```
#include <iostream>
using namespace std;
int main() {
    Student std1, std2;

    std1.setStudentNo(1023);
    std1.assignMarks(50,60,70);

    std2.setStudentNo(2345);
    std2.assignMarks(70,80,75);

    cout <<“Average of student1:”<< std1.calcAvg() << endl;
    cout <<“Average of student2:”<< std2.calcAvg() << endl;
    return 0;
}
```

Activity 3

Implement the Circle class and write a client (main) program to calculate and print the area of a circle.



Reference



Chapter 03

Deitel & Deitel's (2016), C++ How to Program, 9th Edition