

## Chapter 7 - Intro to Classes (7.9 - 7.12)

The class is a C++ construct used to create objects which are fundamental to object-oriented programming (OOP).

- OOP hides the details of objects from objects of other types
- When an object needs information from another object or needs another object to perform a task, it sends a message to the object requesting what it needs
- As a result, object-oriented programs can be written more generically than structured programs
- Usually, making changes to the object-oriented programs is easier than changing structured programs

## class definition

A class is a user-defined datatype that is defined by the programmer. A class consists of variables and functions with a general format as follows:


```
class ClassName  
  
{  
  Declarations for member variables and member  
  functions  
  
};
```

# the person class

```
class Person
```

```
{
```

```
public:
```

```
    int age;  Member Variable
```

```
    int returnAge();
```

```
    int returnBirthYear();
```

```
};  Member Function prototypes
```

```
.....
```

```
int main()
```

```
{
```

```
    Person person;
```

```
    person.age = 28;  Object of class Person
```

```
    cout << "person is: " << person.returnAge();
```

```
    cout << "person was born in: "
```

```
        << person.returnBirthYear();
```

```
    return 0;
```

```
}
```

## person class method definitions

```
int Person::returnAge()  
{  
    return age;  
}
```

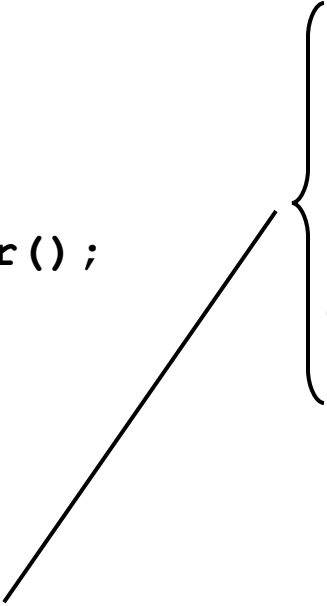
```
int Person::returnBirthYear()  
{  
    return 2010 - age;  
}
```

## private vs public

- Class data members and member functions can be either private or public
- Private data members and member functions can only be accessed within the class in which it is defined
- Public data members and member functions can be accessed from inside or outside of the class in which it is defined

# private vs public example

```
class Person
{
private:
    int age;
public:
    void setAge(int);
    int returnAge();
    int returnBirthYear();
};
.....
int main()
{
    Person person;
    person.setAge(28);
    cout << "person is: " << person.returnAge() << endl;
    cout << "person was born in: "
        << person.returnBirthYear();
    return 0;
}
```



- Because `age` is a private data member, we can't use `person.age = 28` here.
- Instead, we need to create a new function in the class to set the age.

# mutator

A mutator is any method that can change the value of a member variable

```
void Person::setAge(int newAge)
{
    age = newAge;
}
```

An accessor is a method that uses a class member but does not change its value

## a Time class

```
class Time
{
    private:
        int hour;        // 0 - 23 (24-hour clock format)
        int minute;     // 0 - 59
        int second;     // 0 - 59

    public:
        void setTime(int h, int m, int s);
        void printUniversal();           // 13:27:06
        void printStandard();           // 1:27:06 PM
}; // end class Time
```



# class questions

Q1: How many members does class time have? List them.

Q2: How many methods does class time have? List them.

Q3: How many mutators does class time have? List them.

Q4: How many accessors does class time have? List them.

# class questions

P1: Write the definitions of the member functions?

# class questions

Q5: Where would they be written?

Q6: How do we create objects of the class Time?

A regular object

An array of objects

P2: Write C++ code that shows how you would use the objects to call the member functions?

# class questions

P2: Write C++ code that shows how you would use the objects to call the member functions?