

Creating a class

Tutorial 6a

Katja Mankinen

6 October 2017

Aim

The aim of the tutorial is to make you familiar with object oriented programming in C++. Today you will create and use classes!

You should preferably finish all the steps today. It may look like a lot of work, but it is actually very doable! You can immediately start doing the exercises at your own pace. If you have any problems or questions, just ask!

1. Quick Quiz Time

Let's start by answering a short Quiz! Correct answers are at the end of this document but please try to think them before checking the answers.

1. What is the difference between object-oriented programming and procedural programming?
2. What is the difference between public and private data members?
3. What is the `protected` keyword used for?
4. If you declare two Dog objects, can they have different values in their `itsAge` member data?
5. Syntax time: how do you invoke a data member or member function in your program?
6. Syntax time: how do you define a member of a class outside the class?
7. Syntax time: what does `class Apple : public Fruit { /*code*/ };` mean?

2. Students

2.1 Student class

Declare a class called `Student` with these data members: `age`, `GPA`, and student ID.

2.2 Adding accessors

Now when you have your class declared, we can slightly extend it. Make the data members private, and provide public accessor methods to get and set each of the data members.

2.3 Write a program!

Write a program with the `Student` class that makes two students, and sets their age, GPA and student ID. Your program should also print the values (age, GPA, ID). You can choose the values of them yourself.

3. Answers to quiz

1. Procedural programming focuses on functions separate from data. OOP combines data and functions together into objects, and focuses those objects and interaction between them.
2. Public data members can be accessed by everyone who is aware of the class. Private data members can be accessed only by member functions of the class.
3. A protected member of a base class can only be accessed by the members of itself and by the members of classes derived from it.
4. Yes, of course. Each object of a class has its own data members: own ages, names, and so on, depending on the class.
5. Invoking a data member or member function in your program: `instanceName.memberName`. For example Snoopy is an instance of the class Dog, and it has a member `Woof()`. Now we can just write `Snoopy.Woof()`.
6. We define a member of a class outside the class by using the scope operator `::`, for example in our constructor `Dog::Dog(int initialAge)`. Similarly, if we have other functions we want to define outside the class, we can do it by using the same syntax:

```
void ClassName::NameOfMethod()
{
    //do whatever what you want
}
```

7. It means public members in class `Fruit` are inherited by `Apple` and are also public in `Apple`.