C++ Week 4 Constructors

Create a new C++ console application and within it define a new class named GameObject. GameObject should have a single public data member named id of type int.

Exercise 1

Within main declare a static identifier named obj1 of type GameObject and display the value within the id data member within the console.
What value has been displayed and why?
Now ensure the value is initialised to 0 within a default constructor.

Exercise 2

Create a non default constructor that takes the id as an argument and assigns it to the corresponding data member.
Within main create a new static variable named obj2 of type GameObject and invoke the non default constructor by passing it an id of 45.
Now display the id within the console.

Exercise 3

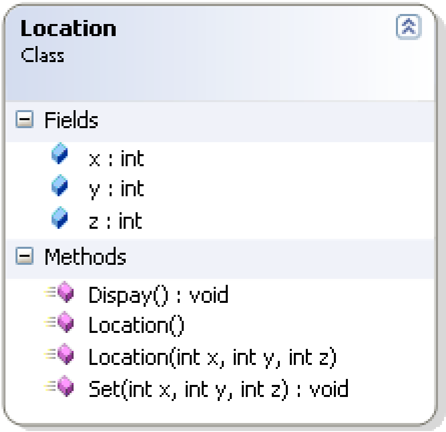
Make the changes necessary to ensure that it is not possible to define an object of the GameObject type using a default constructor. After testing it, comment out the declaration of obj1 and the statement that displays the id.

Exercise 4

Make appropriate changes to GameObject to record how many instances of the object have been created. Create three more static variables of GameObject named obj3,obj4 and obj5 and display the number of instances in the console (it should display 4).

Exercise 5

- Our GameObject objects have a location within the environment recorded by an object of type Location. Add the class to your project.
- Location's Display() outputs the x, y, z values to the console.
- Location's Set() assigns the parameters to the corresponding data members.
- Amend the GameObject's constructor to accept the Location object (as below). Note GameObject now has a data member in which to store the argument passed to the constructor.

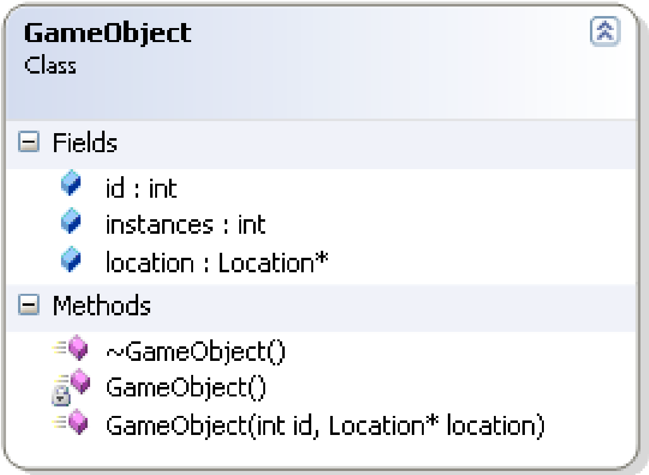


```
Location
Class

Fields
  x : int
  y : int
  z : int
Methods
  Dispay() : void
  Location()
  Location(int x, int y, int z)
  Set(int x, int y, int z) : void
```

C++ Week 4 Constructors

GameObject Class



GameObject
Class

Fields

- id : int
- instances : int
- location : Location*

Methods

- ~GameObject()
- GameObject()
- GameObject(int id, Location* location)

- Within main instantiate an anonymous instance of Location for each of the GameObjects. Each having a unique sequential location starting with (1,1,1) then (2,2,2) etc.
- Within GameObject store the location in the public **pointer location**.
- Invoke each GameObject's location->Display() member function to display the locations within the console.

Exercise 6

- After the statements within main() that display the object locations declare a new static GameObject named obj6 and initialise its value to that of obj4 (**GameObject obj6 = obj4;**).
- Invoke the Display member functions on both (they should both display the same value)
- Now change obj4's location to 8,8,8 using the Set member function.
- Display both locations once more
- Why is obj6's location changed also?
- Correct the problem.

C++ Week 4 Constructors

Exercise 7

- Create a new class called HealthKit that is a derived class (public) of GameObject.
- Delete the HeathKit's default constructor declaration and definition.
- Define a constructor for the new class that accepts the id and location (pointer) arguments and forwards them on to its base class constructor.. The constructor should output the message "**HealthKit Constructore Invoked".**
- Now add a statement that outputs the message **"GameObject Constructor Invoked"** to the non default GameObject constructor.
- Now within main create a static instance of HeathKit passing it an id and anonymous Location.
- Note the order in which the constructors are invoked.