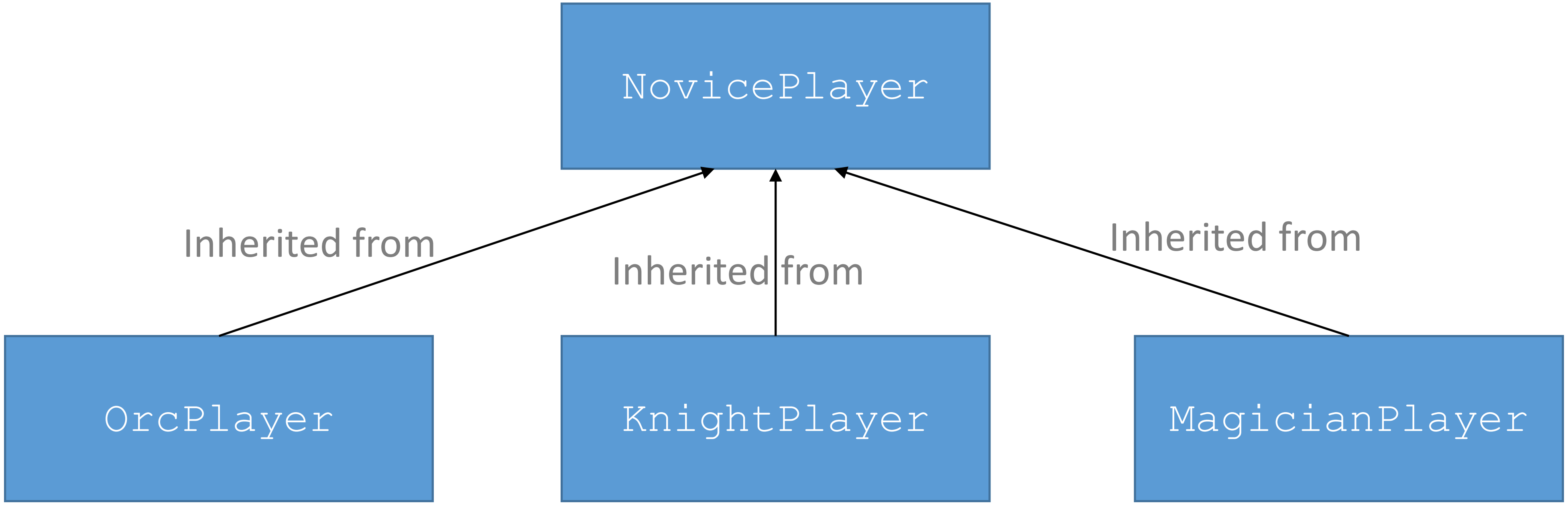
Assignment #7 – Inheritance

Due at next Wed 23:59:59

Introduction to Computers II

Overview

- Implement class "`NovicePlayer`" and its derived classes
- You can add more classes if you'd like to



Data members of `NovicePlayer`

- `private` data members
  - `string name;` // Name of the player
  - `int hp;` // Current HP of the player, range: [0, `max_hp`]
  - `int mp;` // Current MP of the player, range: [0, `max_mp`]
  - `int exp;` // Cumulative experience of the player, >= 0
    // Will not reset to zero after leveling-up
  - `int money;` // Current amount of money that player carries, >= 0

Data members of NovicePlayer

- protected data members
  - int level; // The level of the player, >= 1
  - int attack; // Attack of the player
  - int defense; // Defense of the player
  - int max_hp; // Max HP of the player
  - int max_mp; // Max MP of the player
  - int lvup_exp; // Experience needed for leveling-up

Constructors of `NovicePlayer`

- `NovicePlayer();` // default constructor
  - With `level` = 1, `name` = “anonymous”
- `NovicePlayer(int);` // normal constructor
  - Initializes `level` of the player, `name` = “anonymous”
- `NovicePlayer(int, string);` // normal constructor
  - Initializes `level` and `name` of the player
- `NovicePlayer(const NovicePlayer&);`
  - Copy constructor

Member functions of `NovicePlayer`

- Getter/setters of `name`, `level`, `hp`, `mp`, `exp`, `money`
  - `void setName(string);`
  - `string getName(void) const;`
  - `void setLevel(int);`
  - `int getLevel(void) const;`
  - … (8 other functions left)

- Please note that you should also calculate `attack`, `defense`, `max_hp`, `max_mp` and `lvup_exp` while calling `setLevel()`

Please Notice…

- `hp`, `mp` and `exp` should not greater than `max_hp`, `max_mp` and `lvup_exp`, respectively.
- Please also check other limits on data members and validate values within setters.
- For `attack`, `defense`, `max_hp`, `max_mp` and `lvup_exp` :
  - Should be calculated automatically while initializing and calling `setLevel()`
  - So no general setters for these members
- You can just ignore the procedures while leveling-up, we will deal with it in future classes. Or you can deal with it within `setExp()` if you want.

Member functions of `NovicePlayer`

- Other getters
  - `int getAttack(void) const;`
  - `int getDefense(void) const;`
  - `int getMaxHP(void) const;`
  - `int getMaxMP(void) const;`
  - `int getLvupExp(void) const;`

The `OrcPlayer` class

- Member functions of `OrcPlayer`
  - `OrcPlayer();`
  - `OrcPlayer(int);`
  - `OrcPlayer(int, string);`
  - `OrcPlayer(const OrcPlayer&);`
  - `void setLevel(int);`

The `KnightPlayer` class

- Member functions of `KnightPlayer`
  - `KnightPlayer();`
  - `KnightPlayer(int);`
  - `KnightPlayer(int, string);`
  - `KnightPlayer(const KnightPlayer&);`
  - `void setLevel(int);`
  - `void heal(void);`
- `heal()` does:
  - increasing HP (`level`*10) points by decreasing MP (`level`*5) points

The `MagicianPlayer` class

- Member functions of MagicianPlayer
  - `MagicianPlayer();`
  - `MagicianPlayer(int);`
  - `MagicianPlayer(int, string);`
  - `MagicianPlayer(const MagicianPlayer&);`
  - `void setLevel(int);`
  - `void pray(void);`
- `pray()` does:
  - increasing MP (`level`*10) points by decreasing HP (`level`*5) points

Level-up Formulas

	max_hp	max_mp	attack	defense
NovicePlayer	100 + 10*L	40 + 5*L	20 + 5*L	20 + 5*L
OrcPlayer	200 + 20*L	50 + 5*L	50 + 12*L	30 + 10*L
KnightPlayer	150 + 25*L	70 + 10*L	40 + 10*L	20 + 12*L
MagicianPlayer	120 + 15*L	100 + 15*L	30 + 8*L	20 + 7*L

- "L" stands for "`level`".
- All values can be changed according to your own favor

Experiment Value Formula

- Formula:

$$\texttt{lvup\_exp} = ceiling(10^{\log_2(level+1)})$$

- Example value table

`level`	1	2	3	4	5	6	…
`lvup_exp`	10	39	100	210	385	642	…

Deliverables

- You need to compress all of your class headers and implementations to a zip archive (the naming scheme is same as before)
  - NovicePlayer.h
  - NovicePlayer.cpp
  - OrcPlayer.h
  - OrcPlayer.cpp
  - … (8 files in total)
- Then upload the archive to Moodle