C++ Pool - d09

Koalab koala@epitech.eu

Abstract: This document is the subject for d09

Contents

Chapter I

BASIC RULES

- BASIC RULES :

  - If you do half the exercises because you have comprehension problems, it's okay, it happens. But if you do half the exercises because you're lazy, and leave at 2PM, you WILL have problems. Do NOT tempt the devil.

  - Every function implemented in a header, or unprotected header, means 0 to the exercise.

  - Each class possesses a constructor and a destructor.

  - Every output goes to the standard output and will be ended by a newline, unless specified otherwise.

  - The imposed filenames must be followed TO THE LETTER, as well as class, function and method names.

  - Remember: You're coding in C++ now, and not in C. Therefore, the following functions are FORBIDDEN, and their use will be punished by a -42, no question asked:

    - `*alloc`

    - `*printf`

    - `free`

  - Generally, files associated to a class will always be `CLASS_NAME.h` and `CLASS_NAME.cpp` . (Class name can be in lower case if applicable.)

  - Turn in directories are ex00, ex01, ..., exN

  - Any use of `friend` will result in the grade -42 `no question asked` .

- ○ You must read the examples thoroughly. They can contain requirements that are not obvious in the exercise description.
- ○ You are asked to turn in an important number of classes. However these classes are pretty short. Slackers not accepted!
- ○ Read each exercise FULLY before starting it!
- ○ USE YOUR BRAIN, please!

- EXERCISES COMPILATION:
    - ○ The Koalinette compiles your code with the following flags: `-W -Wall -Werror`
    - ○ In order to avoid compilation issues with the Koalinette, please include the necessary files within the include files (*.hh).
    - ○ Please note that none of your files must contain the `main` function, except if it is explicitly asked. We will use our own `main` to compile and to test your code.
    - ○ You are using the C++ language now. Thus, the compiler is g++!
    - ○ The exercise description can be modified until 4h before the final turn in time! Therefore you must check it regularly!
    - ○ The turn in repository is: `piscine_cpp_d09)/exX` (X being the exercise number).

Chapter II

Exercise -1

Exercise : -1	points : 1
Simple heritage in C, it's easy	
Turn-in directory: `(piscine_cpp_d09)/ex-1`	
Compiler: `gcc`	Compilation flags: `-W -Wall -Werror`
Makefile: `No`	Rules: `n/a`
Files to turn in : `Exo-1.h, Exo-1.c`	
Remarks : `n/a`	
Forbidden functions : `None`	

In order to introduce you to the C++ inheritance, I first give you this short exercise in C. This way, you'll discover what inheritance means, and what it entails when applied to the C language. This will give you a sense of the inheritance implementation under the hood of C++...
It is recommended to be very careful about the output as proposed. You have to deduce your own output and your own display, based on it. This exercise -1 is coded in C, so check your compiler!!!

`By Odin, read the exercise FULLY before you start coding!!!`

Create a structure `s_cthulhu`, that will be `typedef` to `t_cthulhu`. This type is composed of an `int m_power` and a `char* m_name`.

This type is associated to several functions:

- `t_cthulhu* NewCthulhu()` - function that creates a new object `t_cthulhu`, that initializes it and that returns its pointer. When a cthulhu is initialized, the field `m_name` is set to "Cthulhu" and the field `m_power` is set to 42.

- `void PrintPower(t_cthulhu* this)` - function that writes on the standard output the quantity of energy.
- `void Attack(t_cthulhu* this)` - Test if there is enough energy. Minimum energy 42. Consume 42 of energy to do an attack.
- `void Sleeping(t_cthulhu* this)` - Recharges cthulhu and increases its energy by 42000.

Now, create a structure `s_koala` that will be `typedef` to `t_koala` . This type is composed of a `t_cthulhu` called `m_parent` , and a `char m_isALegend` .
Read the previous sentence again.

Read it one more time.

This type is associated to the following functions:

- `t_koala* NewKoala(char* name, char _isALegend)` - function that creates a new object `t_koala` , initializes it and returns it.
- `void Eat(t_koala* this)` - Feed the koala and increase its energy by 42.

Functions :

- `t_cthulhu* NewCthulhu()`
- `t_koala* NewKoala(char *name, char _isALegend)`

Will use the initialization functions:

- `static void KoalaInitializer(t_koala* this, char* _name, char _isALegend)`
- `static void CthulhuInitializer(t_cthulhu* this)`

Now, use the main given in the example to compile your code and display the following output. Yes, that means you have to use your brain!

```
int main()
{
    t_koala* _LKoala = NewKoala(‘‘Legend’’, 1);
    t_koala* _NLKoala = NewKoala(‘‘NotLegend’’, 0);

    t_cthulhu* _cthulhu = NewCthulhu();

    printf(‘‘----Start----\n’’);

    PrintPower(_cthulhu);
    PrintPower(&_LKoala->m_parent);
    PrintPower(&_NLKoala->m_parent);

    Attack(_cthulhu);
    Attack(&_LKoala->m_parent);
    Attack(&_NLKoala->m_parent);

    Eat(_NLKoala);

    Attack(_cthulhu);
    Sleeping(_cthulhu);
    PrintPower(_cthulhu);

    Attack(&_NLKoala->m_parent);

    return (0);
}
```

Expected output:

```
$> ./a.out | cat -e
----$
Building Cthulhu$
Building Legend$
----$
Building Cthulhu$
Building NotLegend$
----$
Building Cthulhu$
----Start----$
Power => 42$
Power => 42$
Power => 0$
Cthulhu attacks and destroys the city$
Legend attacks and destroys the city$
NotLegend can’t attack, he doesn’t have enough power$
NotLegend eats$
Cthulhu can’t attack, he doesn’t have enough power$
Cthulhu sleeps$
```

```
Power => 42000$
NotLegend attacks and destroys the city$
$>
```

Now, it’s time for me to narrate the story of the C++ inheritance ...

Chapter III

Exercise 0

Exercise : 00	points : 2
Every child first needs a parent.	
Turn-in directory: `(piscine_cpp_d09)/ex00`	
Compiler: `g++`	Compilation flags: `-W -Wall -Werror`
Makefile: `No`	Rules: `n/a`
Files to turn in : `Character.hh, Character.cpp`	
Remarks : `n/a`	
Forbidden functions : `None`	

Kreog, a human farmer, shows up at the hero academy with lust for adventure in his eyes. Hero's training requires to succeed at the rite of passage which is generally a quest. Then the trainee specializes to a warrior, a magician etc.

First things first, Kreog is named an apprentice by the general of the Hero Academy, before leaving for his quest. Consequently you have to create a simple class `Character` representing Kreog.

A character possesses a name and a level. The character is constructed only with this name and this level. A character without name nor level is like a pony that is not pink. it's a nonsense

- `Character(const std::string &, int)`
- `const std::string &getName() const`
- `int getLvl() const`
- `int getPv() const`
- `int getPower() const`

Everyone can ask his name and his level to the character.
Whatever happens, a character's life is 100 points maximum, and a character's energy is 100 points maximum. When a character is created, he has the maximum number of life points and of energy points.
It is necessary to be able to know from anywhere the character's number of life points and energy points.

This is Kreog's record.

```
Name        : Kreog
Lvl         : 1
Class       : Character
Race        : Koala

Strength       : 5
Stamina        : 5
Intelligence   : 5
Spirit         : 5
Agility        : 5
```

- Close combat attack : `int CloseAttack()`
  - Cost : 10 of Energy
  - Damage : 10 + Strength
  - Output : "[name] strikes with a wood stick"
- Distant attack: `int RangeAttack()`
  - Cost: 10 of Energy
  - Damage : 5 + Strength
  - Output : "[name] launches a stone"

These two functions return the number of points of damage inflicted by the attack.

- Heal : `void Heal()`

  - Cost : 0 of energy.
  - Cure : Add 50 life points.
  - Output : “[name] takes a potion”

- Restore power : `void RestorePower()`
  - Result : Recharge energy to 100
  - Output : “[name] eats”

A character possesses characteristics such as Stamina, Spirit, and Agility. Theses characteristics are stored as int within the class `Character` .
These characteristics are personalized by children in their constructor.
For example, a `Warrior` possesses 12 point of `Strength` while a `Magician` possesses only 6 `Strength` .

Kreog’s record shows that a basic `Character` possesses a Strength, a Stamina, an Intelligence, a Spirit, and an Agility of 5.

`These characteristics can’t be modified from outside.`

During his first day at the Academy, Kreog learned two pieces of information. The first one is that there are two combat modes: Close combat and distant combat.

Thus, it is possible to specify to the character which combat mode to use during a combat. The following code shows you how:

```
Character _c(‘‘poney’’, 42);

_c.Range = Character::CLOSE;
_c.Range = Character::RANGE;
```

`_c.Range has type AttackRange`

By default the Range is initialized to CLOSE

The second information is that using a technique cost energy. If the character doesn't have enough energy when using a technique, he displays: "[name] out of power" Thus the attack will generate 0 damage.

As strong as he is, a hero can also suffer damages.

- `void TakeDamage(int _damage)`
  - Output : "[name] takes [_damage] damage"

If the character suffers too many damages and if his life points get below 0 (inferior or equal to 0), the character screams: "[name] out of combat".

```
Character _c(‘‘poney’’, 42);

_c.TakeDamage(50);
_c.TakeDamage(200);
_c.TakeDamage(200);
```

```
$>./a.out | cat -e
poney Created$
poney takes 50 damage$
poney out of combat$
poney out of combat$
$>
```

Summary:

```
const std::string& getName() const;
int getLvl() const;
int getPv() const;
int getPower() const;

Character(const std::string&, int)

int CloseAttack();
int RangeAttack();
void Heal();
void RestorePower();
void TakeDamage(int _damage);
```

Chapter IV

Exercise 1

Exercise : 01	points : 3
A children is his/her parent.	
Turn-in directory: `(piscine_cpp_d09)/ex01`	
Compiler: `g++`	Compilation flags: `-W -Werror -Wall`
Makefile: `No`	Rules: `n/a`
Files to turn in : `Character.hh, Character.cpp, Warrior.hh, Warrior.cpp`	
Remarks : `n/a`	
Forbidden functions : `None`	

During the first year, apprentices have to go through a rite of passage which is a quest imposed by the academy so that they prove themselves. Kreog's quest is to recover a particular item in the Tek Dungeon.

The story starts when Kreog walks trough the door of the Tek Dungeon.

After only 10 min in the dungeon, Kreog get crushed by a heavy armor.

```
[Armor] Quountdouce count... By my beard, what on earth did I
stumble on? Ow that's a worm!
[Kreog]  I am not a worm, I am a soon to be hero.

[Armor] Soon to be hero. Ah ah ah. In this case, let me introduce myself.
```

```
Name : Thor
Lvl : 42
Class : Warrior
Race : Dwarf

Strength : 12
Stamina : 12
Intelligence : 6
Spirit : 5
Agility : 7
```

- Close combat attack : `int CloseAttack()`
    - Cost : 30
    - Damages : 20 + Strength
    - Output : "[name] strikes with his [weapon]"
- Weapon : hammer
- Distant attack : `int RangeAttack()`
    - Cost : 10
    - Damages : 0
    - Output : "[name] intercepts"
    - Result: Next attacks will be close combat only
- Soin : `void Heal()`
    - Cost : 0
    - Health :Adds 50 life points
    - Output : "[name] takes a potion"
- Restore power : `void RestorePower()`
    - Result : Recharge energy to 100
    - Output : "[name] eats"
- Creation
    - Output :

“I’m [name] KKKKKKKKKKRRRRRRRRRRRRRREEEEEEEEOOOOOORRRRGGGGGGG”

The type of the weapon is initialized in the constructor !

overload

Create a class `Warrior` that inherits from the class `Character` . In the end a Warrior `IS` a character. As with his parent, a warrior is constructed only with a name and a level.

A warrior’s characteristics are different from a basic character.

Moreover, a warrior chooses his weapon when being constructed. Therefore, the `Warrior` class possesses a field `std::string weaponName` .

Warrior are very proud beings. Therefore, they don’t allow anyone else to change the weapon they are using.

```
[thor]     When a warrior is born, he makes his first scream.
           For me it was lordly! I still remember it:

I'm Thor KKKKKKKKKKRRRRRRRRRRRRRREEEEEEEEOOOOOORRRRGGGGGGG

[Thor]     So, young novice, did you finish your rite of passage?

[Kreog]    No, I just arrived!

[Thor]     Perfect, I am gonna come with you and help you in your quest!
```

Chapter V

Exercise 2

	Exercise : 02 points : 4
Children, always children	
Turn-in directory: `(piscine_cpp_d09)/ex02`	
Compiler: `g++`	Compilation flags: `-W -Werror -Wall`
Makefile: `No`	Rules: `n/a`
Files to turn in : `Character.hh, Character.cpp, Warrior.hh, Warrior.cpp, Mage.hh, Mage.cpp, Priest.hh, Priest.cpp`	
Remarks : `n/a`	
Forbidden functions : `None`	

Thor is now part of Kreog team. Traveling through the Dungeon, and while Thor was babbling again and again, a fire ball passed them very closely, burning some of Thor's beard. A very angry Thor starts looking at the origin of the fire ball, when a second one razed his feet. While looking down, he saw a gnome running in every direction. Suddenly this one froze, stupefied by a magical spell that a nearby Goblin cast!

Coming out of nowhere, a light surrounded the gnome and freed him. Growing more and more angry by all this racket, Thor charged the Goblin and hit him with his powerful hammer. The gobelin fled over the room and crashed violently on a pile of rocks at the other side of the room.

```
[Kreog] How are you little being??

[Gnome] Little being??? Did you look at yourself?
I am one of the tallest among my people!
Let me introduce myself:

Name         : Sully
Lvl          : 40
Class        : Mage
Race         : Gnome

Strength          : 6
Stamina           : 6
```

```
Intelligence     : 12
Spirit           : 11
Agility          : 7
```

- Close combat attack :
  - Cost : 10
  - Damage: 0
  - Output : “[name] blinks”
  - Result : After this attack the Magician will use a distant attack.
- Distant attack :
  - Cost : 25
  - Damages : 20 + Spirit
  - Output : “[name] launches a fire ball”
- Heal :
  - Cost : 0
  - Cure : Adds 50 life points
  - Output : “[name] takes a potion”
- Restore power :
  - Result : Recharge 50 energy points + Intelligence
  - Output : “[name] takes a mana potion”
- Creation :
  - Output : “[name] teleported”

To create a Mage object:

```
Mage(const std::string& name, int lvl)
```

```
[Sully] Let me introduce my companion.

Name        : Iopy
Lvl         : 84
Class       : Priest
Race        : Orc

Strength     : 4
Stamina      : 4
Intelligence : 42
Spirit       : 21
Agility      : 2
```

- Close combat attack :
  - Cost : 10
  - Damages : 10 + Spirit
  - Output : “[name] uses a spirit explosion”
- Distant attack :
  - Cost : 25
  - Damages : 20 + Spirit
  - Output : “[name] launches a fire ball”
- Heal :
  - Cost : 10
  - Cure : Adds 70 life points
  - Output : “[name] casts a little heal spell”
- Creation :

- Output : “[name] enters in the order”

A priest is a magician specialized in sacred magic.

Chapter VI

Exercise 3

Exercise : 03	points : 4
A paladin is a being mixing priest and warrior characteristics	
Turn-in directory: (piscine_cpp_d09)/ex03	
Compiler: g++	Compilation flags: -W -Werror -Wall
Makefile: No	Rules: n/a
Files to turn in : Character.hh, Character.cpp, Warrior.hh, Warrior.cpp, Mage.hh, Mage.cpp Priest.hh, Priest.cpp, Paladin.hh, Paladin.cpp	
Remarks : n/a	
Forbidden functions : None	

After these introductions, Sully and Iopi decided to escort Thor and Kreog in the quest. After few hours of wandering and fighting, the group finds a room with a strange man, covered in sweat, and dancing in the middle of the room.

```
[Thor] Bibi ! Is that you ???
[The sweating man] Thor, long time no see my friend!

Name : Bibi
Lvl : 42
Class : Paladin
Race : Humain

Strength : 9
Stamina : 10
Intelligence : 10
Spirit : 10
Agility : 2
```

- Close combat attack :

  - Cost : 30
  - Damages : 20 + Strength
  - Output : “[name] strikes with his [weapon]”

- `Weapon : hammer`
- Distant attack :
  - Cost : 25
  - Damages : 20 + Spirit
  - Output : “[name] launches a fire ball”
- Heal :
  - Cost : 10
  - Cure : Adds 70 life points
  - Output : “[name] casts a little heal spell”
- Restore power :
  - Result : Recharges energy to 100
  - Output : “[name] eats”
- Creation :
  - Output : “the light falls on [name]”

`A Paladin is a mix between a Warrior and a Priest.`

For the inheritance order, follows the order as presented previously.

The paladin uses the priest healing spell, the priest fire ball spell and the close combat attack of the Warrior. Moreover, he can charge as a warrior.

```
int Intercept()
```

Warrior description & using

Note : During virtual inheritance, the class inherited virtually is created first. Thus, this class is initialized in first too.

Bibi, still sweating gave a big hug to his lifelong friend Thor. After hours of fairy tales and old souvenirs, the team, half asleep, decided to accept Bibi as a new group member for the quest.

Chapter VII

Exercise 4

Exercise : 04	points : 6
From leaf to tree, an elf is a recurring character.	
Turn-in directory: (piscine_cpp_d09)/ex04	
Compiler: g++	Compilation flags: -W -Werror -Wall
Makefile: No	Rules: n/a
Files to turn in : Character.hh, Character.cpp, Warrior.hh, Warrior.cpp, Mage.hh, Mage.cpp, Priest.hh, Priest.cpp, Paladin.hh, Paladin.cpp, Hunter.hh, Hunter.cpp	
Remarks : n/a	
Forbidden functions : None	

Upset because of Bibi and Thor's old memories, the other team members decided to charge a pack of Goblins to wash theirs heads. Suddenly, an arrow flying across the room strikes the Gobelin right in front of Kreog.
Surprised, Kreog turned around and saw a little green something jumping from one side of the room to the other. The fight was over, and a very cute she-elf emerged from the shadow to get close to the group.

```
[Thor]  Quountdouce count... a she-elf in the dungeon!
        You are pretty far from your home-wood, little miss!

[Elf] LITTLE MISS???  Your eyes are too deep into your
beard to see correctly dirty dwarf. I am NOT A MISS I am an HE-ELF!

Name         : Ben
Lvl          : 40
Class        : Hunter
Race         : Elf

Strength      : 9
Stamina       : 9
Intelligence : 5
Spirit        : 6
```

```
Agility      : 24
```

- Close combat attack :
  - Cost : 30
  - Damages : 20 + Strength
  - Output : “[name] strikes with his [weapon]”
- Weapon : sword
- Distant attack :
  - Cost : 25
  - Damages : 20 + Agility
  - Output : “[name] uses his bow”
- Heal :
  - Cost : 0
  - Cure : Adds 50 life points
  - Output : “[name] takes a potion”
- Restore power :
  - Result : Recharge energy to 100
  - Output : “[name] meditates”
- Creation :
  - Output : “[name] is born from a tree”

A hunter is a character that uses warrior characteristics for the close combat. However, the elf doesn't want to be considered a warrior by the other members of the group. Even if it is a family tradition to be associated to a warrior within the clan.

Chapter VIII

Conclusion

After hours of arguments with the dwarf, Ben joined the group and they moved on through the dungeon. Sadly, Sully asked Iopy who is the strongest between the Dwarf and the Elf. This question started a violent debate, making way more noise than a Hobbit in the Moria.

The argument was so noisy that a dragon got into the room to check what was going on..
[Dragon] Hey, can you shut up now? I am trying to sleep here!

[Thor] Where the hell is this guy coming from. Can't he leaves us to our friendly debate? Give me 2 minutes and I knock him out.

[Kreog] STOOOOOOOOOP !!!! Hey in my quest description, a sacred dragon is mentioned! Dragon, would you have something for me!?
[Dragon] I am ok to give you my treasure but I have two conditions:

- First thing first, you tell the dwarf and the elf to shut up their large mouths.
- Secondly, you tell the academic big guys to use another dungeon to test theirs little punks. I am tired of these soon to be losers!

Kreog accepted the deal and asked the dwarf and the elf to be quiet. Five minutes later the dragon brought back a stuffed Koala and handed it to Kreog.
[Dragon] Here it is the glorious object of your quest!.
[Thor] What the F... I had to strike a deal with an Elf to get a stuffed Koala? Quount-douce count... This is the last time I am teaming with newbies ...

The quest being over, the group gets back to the academy and organized a huge party.
Ok guys, I know the end is a little bit short, but if you can find something better, just post it on the forum. If one of them is worth it, it will be included in next year's exercise!