Homework 02

[2023-1] 데이터사이언스를 위한 컴퓨팅 1 (001)

Due: 2023 년 10 월 19 일 목요일 23 시 59 분

1. C++ Polymorphism: Shape [50pts]

As shown in Figure 1, the `main` function is given.

```cpp
#include <cstdlib>
#include <iostream>
#include "mycode.h"

int main() {
    Square sq(5.0f);
    Equilateral eq(4.0f);

    sq.print();
    eq.print();

    Shape *s1 = &sq;
    Shape *s2 = &eq;

    s1->print();
    s2->print();

    return EXIT_SUCCESS;
}
```

Figure1. `main` function

Write code that defines a C++ class and virtual member so that when the given `main` function is executed, the result is as shown in Figure 2.

```
This is Square. Area is 25.000000
This is Equilateral. Area is 6.928203
This is Shape. Area is 25.000000
This is Shape. Area is 6.928203
```

Figure2. output `stdout` when running main function

Instruction: Write code to satisfy the following conditions.

1. The `Shape` class receives one single-precision floating-point number as an argument to the constructor. This argument is the length of one side of the shape.
2. The `Shape` class has a virtual function called `area` that returns a single-precision floating-point type.
3. The value returned from the `area` function is the area value of the shape calculated using

the length value of one side of the shape.

4. `Square` class and `Equilateral` class inherit `Shape` class.
5. The `print` function uses the previously defined `area` function to output a string as shown in the example in Figure 2.
6. DO NOT modify the given `main.cpp` file and write all necessary code in the `mycode.cpp` and `mycode.h` files.

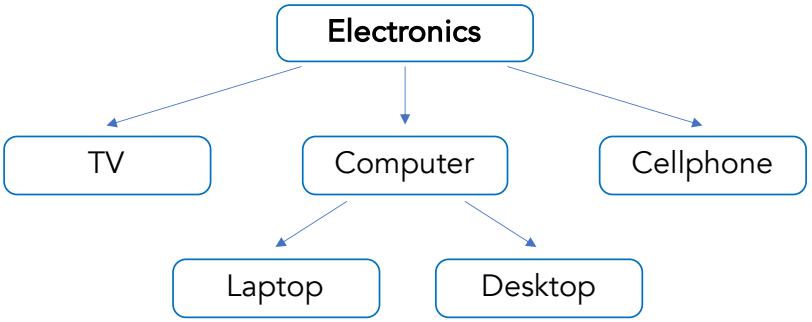
`Makefile` is provided to help compilation and job scheduling. Please compile and run it at GSDS server.

```
cfdsta@login0:~/hw2/Shape$ make clean
rm -rf oop main.o mycode.o
cfdsta@login0:~/hw2/Shape$ make
g++    -c -o main.o main.cpp
g++    -c -o mycode.o mycode.cpp
g++ -o oop main.o mycode.o
cfdsta@login0:~/hw2/Shape$ make test
salloc --partition=class4 --nodes=1 --ntasks-per-node=1 --time=5 --cpus-per-task=1 --mem=1G ./oop
salloc: Granted job allocation 26564
This is Square. Area is 25.000000
This is Equilateral. Area is 6.928203
This is Shape. Area is 25.000000
This is Shape. Area is 6.928203
salloc: Relinquishing job allocation 26564
```

2. Electronics [50pts, NO partial credit]

Consider three different types of Electronics, TV, Computer and Cellphone. TV is the heaviest, and then Computer and then Cellphone.

Instruction:



1. The price is **size** x **w**, where **w = 1.8, 0.6, 0.7** for TV, Computer and Cellphone, respectively.
2. Consider a Laptop, which can have one GPU as max. When there are GPU, the price becomes twice.
3. Consider a Desktop, which can have two GPU as max. When there are GPU, the price is 1.8 times higher.
4. Total price of Desktop can be calculated as the sum of the price of Desktop and GPUs.
5. The `Electronics` class have
    a. `Size (member variable)`
    b. `SetSize`
    c. `GetSize`
6. Additionally, have
    a. `void PrintSelf()`
    b. `void AddGPU(Electronics *aGPU)` (for `Desktop` and `Laptop`)
7. `PrintSelf` prints

    `It is a {Electrotic type}, Size={size}, Price={price}`

    or (in the case of Desktop)

    `It is a {Electrocis type} with { # of GPUs} GPU(s), Size={size}, Price={price}, Total Price = {totalprice}`

**** For Laptops, DO NOT override Computer's `PrintSelf`.***

**** You can include the necessary C++ standard library header files and define additional functions.***

**** Please omit all digits after the decimal point and print it out as shown in the example below, paying attention to spacing and capital letters.***

```
cfdsta@login0:~/hw2/Electronics$ make clean
rm -rf Electronics main.o Electronics.o
cfdsta@login0:~/hw2/Electronics$ make
g++    -c -o main.o main.cpp
g++    -c -o Electronics.o Electronics.cpp
g++ -o Electronics main.o Electronics.o
cfdsta@login0:~/hw2/Electronics$ make test
salloc --partition=class4 --nodes=1 --ntasks-per-node=1 --time=5 --cpus-per-task=1 --mem=1G ./Electronics
salloc: Pending job allocation 26567
salloc: job 26567 queued and waiting for resources
salloc: job 26567 has been allocated resources
salloc: Granted job allocation 26567
It is a TV, Size=600, Price=1080
It is a Cellphone, Size=160, Price=112
It is a Computer, Size=330, Price=396
It is a Desktop with 2 GPU(s), Size=400, Price=432, Total Price=474
salloc: Relinquishing job allocation 26567
```

3. Submission Instruction

- **You cannot change `main.cpp` and `Makefile`**. You don't need to submit them.
- **Any disadvantages caused by not referring to the submission instructions will be scored without exception.**
- Make sure your code works well on the GSDS server. Your code will be scored automatically by the program on the GSDS server. If you don't follow the submission instruction, a penalty may occur.
- If you want to use your grace day, you must notify the TA by e-mail when submitting the homework. If you don't notify, we will judge that you want to save your grace day for the next homeworks, so your homework is considered unsubmitted. Even if you use your grace day, your homework should be submitted through the submission command.
- Copy the skeleton code from /cfds/skeleton/hw2

```
cfds100@login0:~$ cp -r /cfds/skeleton/hw2 .
```

- Submission code guide
  cfds-submit[status|submit|diff|cat|pull][hw1|hw2|...]filename

  status: 과제 현황 보기

```
cfds100@login0:~/hw2$ cfds-submit status hw2
hw2
  mycode.h          : NOT SUBMITTED
  mycode.cpp        : NOT SUBMITTED
  Electronics.cpp   : NOT SUBMITTED
  Electronics.h     : NOT SUBMITTED
```

  The submission method is the same as hw1.