Week 7-1 : Virtual Function & Polymorphism

Part1. Virtual Function

Keyword of the virtual function : virutal

not define a function's body

When virtual function is defined, overriding this function is also virtual function.

prevent to generate a wrong object and state clearly that function does not executing actually.

```
#include <iostream>
using namespace std;

class First
{
public:
        void MyFunc() { cout<<"FirstFunc"<<endl; }
};

class Second: public First
{
public:
        void MyFunc() { cout<<"SecondFunc"<<endl; }
};

class Third: public Second
{
public:
        void MyFunc() { cout<<"ThirdFunc"<<endl; }
};

int main(void)
```

```
{
        Third * tptr=new Third();
        Second * sptr=tptr;
        First * fptr=sptr;

        fptr->MyFunc();
        sptr->MyFunc();
        tptr->MyFunc();
        delete tptr;
        return 0;
}
```

Print out FirstFunc, SecondFunc, ThirdFunc. This is not what programmer intend. so change this code.

```
#include <iostream>
using namespace std;

class First
{
public:
        virtual void MyFunc() { cout<<"FirstFunc"<<endl; }
};

class Second: public First
{
public:
        virtual void MyFunc() { cout<<"SecondFunc"<<endl; }
};

class Third: public Second
{
public:
        virtual void MyFunc() { cout<<"ThirdFunc"<<endl; }
};

int main(void)
{
        Third * tptr=new Third();
```

```
        Second * sptr=tptr;
        First * fptr=sptr;

        fptr->MyFunc();
        sptr->MyFunc();
        tptr->MyFunc();
        delete tptr;
        return 0;
}
```

Result is printing out 'ThirdFunc' three time. Not to decide called target based on pointer data type but to decide called target based on referencing object pointed by pointer variable.

- Pure Virtual Function

```
class Employee
{
private:
        char name[100];
public:
        Employee(char * name) { .... }
        void ShowYourName() const { .... }
        virtual int GetPay() const
        {
        return 0;
        }
        virtual void ShowSalaryInfo() const
        { }
};
```

Base class for inheritance(Class which goal is not to generate object). To prevent a wrong grammar.

- Use of pure virtual function

```
class Employee
{
private:
```

```
        char name[100];
public:
        Employee(char * name) { .... }
        void ShowYourName() const { .... }
        virtual int GetPay() const = 0;
        virtual void ShowSalaryInfo() const = 0;
};
```

For notifying not to define function's body explicitly to a compiler, write 'const = 0'.

[Exercise]

Using a pure virtual function, Implement dog and cat which inherits Animal Class and make a roar() function. Show the polymorphism in the main function.

- Animal class have a private name variable.
- Dog and Cat class have a private id variable.

hint : when you initialize the Dog and Cat object, use an Animal class