**Date: 26/Jan/2021**

Week4

Chapter2: **Classes and Objects**

1. Classes: ជាពុម្ពគម្រូសំរាប់បង្កើត Objective

**\*Syntax:**

**class** className{

**private/protected/public:**

data members; // (member variable) (Field)

**public:**

function member; (member function)

……………………..

};

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date: 02/Feb/2021**

**Data** members, Field

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date: 04/Feb/2021**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: 11/Feb/2021**

**::** is a scope operator

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date: 16/Feb/2021**

**4. Constructors:**

ជា Function member(Mananger Function) ដែលមានមុខងារសំរាប់ផ្តើមតំលៃអោយObjects។ហើយវាមានឈ្មោះដូចClassនិងគ្មាន Return Typeទេ ដែលចែកចេញជាពី

- **Default constructor**: សំរាប់តំលៃObjectដោយមិនបញ្ជូន Data ពីនៅកន្លែងបង្កើតObjectsឡើយ។

- **Constructor with parameter**: សំរាប់តំលៃObjectដោយបញ្ជូន Data ពីនៅកន្លែងបង្កើតObjectsផ្ទាល់។

- **Copy constructor**: សំរាប់តំលៃObjectដោយចំលង Data ពី Object ណាដែលមានស្រាប់

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date: 16/Feb/2021**

**Trim or space for display on console:**

**cout.width(20); cout.fill(“ “); cout.setf(ios::left);**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date: 25/Feb/2021**

**6. Inner class**

ការបង្កើត class មួយ វាស្ថិតនៅក្នុងclassផ្សេងមួយទៀត។ class ខាងក្រៅOuter class ឬ class ខាងក្នុងគេហៅថា Inner class។

**+ Syntax នៃការបង្កើត Nested class:**

class outer className{

class Inner className{

}

...........................

...........................

}

Ex: បង្កើត class Point (ជា Inner class) តាងអោយចំណុចហើយបង្កើត class circle(ជា Outer class) តាងអោយរង្វង់

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date: 04/Mar/2021**

**7. Dynamic Memory Allocation:**

បង្កើត Memory space ក្នុងពេលRun សំរាប់ផ្ទុកData។ប្រើប្រមាណវិធី new ហើយ deallocate វិញ delete

delete p;

example: delete[] a;

**8. Pointer this:**

ជាPointer ដែលកើតឡើងដោយស្វ័យប្រវត្តិនៅពេល objectកើតឡើងហើយវា Point ទៅកាន់ objectនោះ។ គេមាន Access វាបានតែក្នុង Definition របស់ Function តែប៉ុណ្ណោះ។

Ex: **class** ClassThis{

**private**:

int x;

**public**:

ClassThis() { **this->** x=10; }

ClassThis(int a) { **this->** x=10; }

void input() { cout<<”input x”; cin>>**this-**>x; }

void output() { cout<<”x= ”<< **this->**x<<**endl**;

}

**8. Destructor:**

ជាFunction member (Manager function) វាអនុវត្តជាស្វ័យប្រវត្តិនៅពេល object បាត់បង់។វាមានឈ្មោះដូច class គ្មាន returnType ទេហើយនាំមុខដោយ សញ្ញា ~ (tide)។

**\*Syntax:**

**~**className(){..................}

Ex:..................

class Sample{

private:

.......................

public:

Sample(); // Default constructor

**~**Sample(){ ......................... } // Destructor

.........................

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date: 04/Mar/2021**

**9. Inline:**

**inline** function use for make function execute fast than without inline

**11. Static class Member:**

ជា members របស់ class ដែលមិនមែនជា member របស់ objectទេ(មិន មែនកម្មសិទ្ធរបស់objects)។ ហើយមាន២ប្រភេទគឺ:

a- Static data members

ជាdataរបស់classដែលមានលក្ខណះរួមទៅគ្រប់objectទាំងអស់

**Note:** static មិនអាចបញ្ចូលពី Keyboardទេ

**Ex:** បើកម្មករទាំងអស់ទទូលកំរៃ 3.5$ ក្នុង១ម៉ោង

class Worker{

**private:**

**................................**

**static float** rate:

................................

};

**float** Worker::rate=3.5;

b- Static Function Members

ជាmember របស់ classតែមិនមែនជាកម្មសិទ្ធសំរាប់ដោះស្រាយ លើobjectsទាំងអស់រួមគ្នា។

\*Syntax:

**static** returnType funcName(Parameters);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date: 16/Mar/2021**

**12. Friend Function:**

ជាNon function Member ដែលអាចAccess គ្រប់ members(ដែល ជាPrivate) តាមរយះ object ណាមួយ។

\*Syntax:

friend rerturnType funcName(parameters);

**12. Friend Class:**

Class ដែលផ្តល់friendshipគឺclass ណាមួយ ហៅថា friend class។ ប្រយោជន៍គឺវាអាច access បានគ្រប់members(ដែលជា private) របស់ class ដែលផ្តល់ friendship ទៅគេ។

Ex: class A វាបានផ្តល់ អោយ class B នោះ B អាច access លើ A បាន តែ A មិនអាច access លើ B វិញបានទេ។

\*Syntax:

class A{

friend class B;

.......................

}

class B{

..........................

..........................

}

**14. Common friend function:**

ជា non function member មួយដែលមានfriendshipជាមួយclass ច្រើន។ គេត្រូវប្រកាស friend function នោះនៅគ្រប់ class scope ទាំងអស់ ហើយ definition របស់វាកំណត់ឡើងតែមួយគត់នៅក្នុង class ចុងក្រោយឬ នៅ ក្រៅclass នៃគ្រប់ class ទាំងអស់។ ហើយ class ទាំងអស់ត្រូវប្រកាសជា លក្ខណះ forwad។

Ex: .......................