N211FR

இனக்குழுக்கள் மற்றும் பொருள்கள்

கற்றலின் நோக்கங்கள் :

இந்தப்பாடத்தைக் கற்றபின், மாணவர்கள்

- இனக்குழு, பொருள்கள், ஆக்கிகள் மற்றும் அழிப்பியின் பயன்பாடுகளை புரிந்து கொள்ளுதல்.
- ஆக்கி மற்றும் அழிப்பியுடன் கூடிய இனக்குழுவை பயன்படுத்தி C++ நிரல் எழுதுதல்.
- ஆக்கி மற்றும் அழிப்பியுடன் கூடிய C++ நிரலை இயக்கி தவறுகளை திருத்துதல்.

14.1 இனக்குழு அறிமுகம் :

C++ ன் மிக முக்கியமான பண்புக்கூறு இனக்குழு என்பதாகும். இது ஜேர்ன் ஸ்ட்ரௌஸ்ட்ரப் (Bjarne Stroustrup) என்பவரால் உருவாக்கப்பட்டது, முதலில் அவர் சூட்டிய பெயர் "C WITH CLASSES" என்பதாகும். C++ ஆனது, பொருள்நோக்கு நிரலாக்க மொழியின் பொதுவான நான்கு அடிப்படை அம்சங்களை அதாவது அருவமாக்கம், உறைபொதியாக்கம், மரபுரிமம் மற்றும் பல்லுருவாக்கம் ஆகியவற்றை அளிக்கிறது.

14.1.1 இனக்குழுவின் தேவை:

இனக்குழுவானது தரவுகளையும் அவை தொடர்பான செயல் கூறுகளையும் ஒன்றாக சேர்த்துவைக்க வழிசெய்கிறது. இனக்குழுவானது தரவுகளையும், அவற்றிற்கான பண்புகளையும் மட்டும் கொண்டிருக்காமல், நடப்புலகில் நிலவும் பொருளைக் குறிப்பிட உதவுகிறது. மேலும் அப்பொருள் தொடர்பான செயல்பாடுகளையும் கொண்டிருக்கும். இது பயனர் வரையறுக்கும் தரவினத்தை உருவாக்கப் பயன்படுகிறது.

14.1.2 இனக்குழு வரையறை (Declaration of a class)

C++ல் இனக்குழுவின் வரையறையில் Class என்னும் சிறப்புசொல்லைத் (Keywords) தொடர்ந்து இனக்குழுவின் பெயர் இடம்பெற வேண்டும். இனக்குழுவின் உடற்பகுதி வரையறுப்பானது நெளிவு அல்லது கொக்கி வடிவ அடைப்பு ({ }) குறிக்குள் அடைக்கப்பட்டு அரைப்புள்ளியுடன் அல்லது இனக்குழு பொருள்கள் அறிவிப்புடன் முடிவடைதல் வேண்டும்.

குறிப்பு

கட்டுரு மற்றும் இனக்குழுவிற்கு இடையேயான வேறுபாடானது, கட்டுரு உறுப்புகளானது கொடாநிலையாக Public அணுகியல்புடனும் இனக்குழுவின் உறுப்புகளானது Private அணுகியல்புடனும் இருக்கும்.

இனக்குழுவை வரையறுப்பதற்கான பொது வடிவம் :

```
class class.name
      {
      private :
      தரவு உறுப்புகள் அறிவிப்பு ;
      செயல் கூறுகள் அறிவிப்பு ;
      protected :
      தரவு உறுப்புகள் அறிவிப்பு ;
      செயல் கூறுகள் அறிவிப்பு ;
      public :
      தரவு உறுப்புகள் அறிவிப்பு ;
      செயற்கூறுகள் அறிவிப்பு ;
      };
```

- இனக்குழுவின் உடற்பகுதியானது தரவு உறுப்புகளின் அறிவிப்பை கொண்டிருக்கும். (தரவு உறுப்புகள் மற்றும் உறுப்பு செயற்கூறுகள்)
- இனக்குழுவின் உடற்பகுதியானது மூன்று அணுகியல்பு வரையறுப்பிகளை கொண்டுள்ளது. (Private, Protected மற்றும் Public)

Chapter 14.indd 233 4/11/2020 11:49:09 AM

14.1.3 இனக்குழு அணுகியல்பு வரையறுப்பிகள் :

பொருள்நோக்கு நிரலாக்க மொழியின் மிக முக்கியமான பண்புக்கூறுகளுள் ஒன்று தரவு மறைப்பு (Data hiding) என்பதாகும். இது நிரலில் உள்ள செயற்கூறானது, இனக்குழுவிற்குள் அறிவிக்கப்பட்டிருக்கும் உறுப்புகளை (தரவு உறுப்புகள், உறுப்புச் செயல் கூறுகள்) அணுகுவதற்குத் தடை விதிக்கிறது. இனக்குழுவின் உறுப்புகளை அணுக தடைவிதிப்பதற்கு இனக்குழுவினுள் அறிவிக்கப்பட்டிருக்கும் Private, Protected மற்றும் Public என்ற சிறப்புச் சொற்கள் பயன்படுகிறது.

Private, Protected மற்றும் Public என்ற சிறப்புச் சொற்களானது அணுகியல்பு வரையறுப்பிகள் என அழைக்கப்படுகின்றன. இனக்குழு உறுப்புகளின் கொடாநிலை அணுகியல்பானது Private ஆகும்.

Public உறுப்புகள் :

Public அணுகியல்புடன் கூடிய உறுப்புகளை இனக்குழுவிற்கு வெளியில் இருந்தும் அணுக முடியும். Public என அறிவிக்கப்பட்ட தரவு உறுப்புகளை எவ்வித உறுப்பு செயல்பாடுகளின் உதவியுமின்றி அணுக முடியும். அதாவது தரவு உறுப்புகளுக்கு மதிப்பைக் கொடுக்கவோ அல்லது மதிப்பைப் பெறவோ முடியும். உறுப்பு செயற்கூறுகள் தேவை இல்லை.

Private உறுப்புகள் :

Private அணுகியல்புடன் கூடிய இனக்குழு உறுப்புகளை இனக்குழுவிற்கு வெளியில் இருந்து அணுகமுடியாது. இனக்குழுவின் உறுப்பு செயற்கூறுகள் மட்டுமே தரவு உறுப்புகளை அணுக முடியும். கொடாநிலையாக இனக்குழுவின் உறுப்புகள் அனைத்தும் Private அணுகியல்புடன் அறிவிக்கப்படும்.

Protected உறுப்புகள் :

Protected அணுகியல்புடன் அறிவிக்கப்பட்ட தரவு உறுப்புகளும், உறுப்பு செயற்கூறுகளும் Private அணுகியல்பு கொண்ட உறுப்புகளைப் போலவே செயல்படும். ஆனால் அந்த இனக்குழுவை அடிப்படையாகக் கொண்டு தருவிக்கப்பட்ட இனக்குழுவிற்குள்ளும் அணுக முடியும்.

எடுத்துக்காட்டு

class என்ற சிறப்புச் சொல்லானது இது ஒரு இனக்குழு வரையறை என்பதை நிரல் பெயர்ப்பிக்கு உணர்த்துகிறது

இனக்குழுவின் பெயரானது பயனர் வரையறுக்கும் தரவினத்தின் பெயராகும். இதை பயன்படுத்தி அந்த இனக்குழுவில் பொருள்களை உருவாக்க முடியும்.

```
class student
{
private:
        char name [10];
        int rollno, mark1, mark2, total;
protected:
        void accept();
        void compute();
public:
        void display();
};
```

இவை *private* அணுகியலுடன் கூடிய தரவு உறுப்புகளாகும் அதாவது, இவ்வுறுப்புகளை இனக்குழுவிற்கு வெளியிலிருந்து அணுகமுடியாது.

இவை *protected* அணுகியலுடன் கூடிய உறுப்பு செயற்கூறுகளாகும் அதாவது, இவ்வுறுப்புகளை இனக்குழுவிற்கு வெளியிலிருந்து அணுகமுடியாது. ஆனால் தருவிக்கப்பட்ட இனக்குழுவிற்குள் அணுகமுடியும்.

இவை *public* அணுகியலுடன் கூடிய உறுப்பு செயற்கூறுகளாகும். இவ்வுறுப்புகளை இனக்குழுவிற்கு வெளியிலிருந்து அணுகமுடியும்.

குறிப்பு

ஓர் இனக்குழுவில் உள்ள அனைத்து உறுப்புகளும் (தரவு உறுப்புகள், உறுப்பு செயல் கூறுகள்) Private அணுகியல்புடன் அறிவிக்கப்பட்டால் அந்த இனக்குழுவானது செயலற்றுப் போகும். இனக்குழுவின் பொருளானது இனக்குழுவின் உறுப்புகள் எதையும் அணுக முடியாது.

செயல்பாடு 1 :

பின்வரும் குறிமுறையில் பிழைக்கான காரணம் கூறு

(i)	(ii)

class count
{
int first;
int second;
public:
int first;
};
	class item
	{
	int prd;
	};
	item int prdno;

14.1.4 இனக்குழு உறுப்புகளின் வரையறை

இனக்குழுவானது உறுப்புகளை உள்ளடக்கியதாகும். உறுப்புகளானது தரவு உறுப்புகள் மற்றும் உறுப்பு செயற்கூறுகள் என வகைப்படுத்தப்படும். தரவு உறுப்புகள் என்பவை தரவு மாறிகள் எனப்படும். இவை இனக்குழுவின் பண்புக்கூறுகளைக் குறிப்பதாகும். உறுப்பு செயற்கூறுகள் என்பவை ஓர் இனக்குழுவானது குறிப்பிட்ட செயல்பாட்டைச் செய்ய உதவும் செயற்கூறுகளாகும். உறுப்பு செயற்கூறுகளானது வழிமுறைகள் (Methods) எனவும் தரவு உறுப்புகளானது பண்புக்கூறுகள் (attributes) எனவும் அழைக்கப்படும்.

குறிப்பு :

இனக்குழுவானது ஆக்கி மற்றும் அழிப்பி எனப்படும் தனிச்சிறப்பான உறுப்பு செயற்கூறுகளைக் கொண்டிருக்கும்.

எடுத்துக்காட்டு

```
class result
{
private;
        char name [10];
        int rollno,mark1, mark2, total;

public:
        void accept();
        void display();
};
```

தரவு உறுப்புகள்

உறுப்பு செயற்கூறுகள்

14.1.5 உறுப்பு செயற்கூறுகளை வரையறுத்தல் :

இனக்குழுவின் உறுப்புச் செயற்கூறுகளை வரையறுக்காமல் இனக்குழுவின் அறிவிப்பானது முழுமையடையாது. இனக்குழுவின் உறுப்பு செயற்கூறானது இரண்டு விதங்களில் வரையறுக்கப்படுகிறது.

1) இனக்குழுவிற்கு உள்ளே வரையறுத்தல்

2) இனக்குழுவிற்கு வெளியே வரையறுத்தல்

1) இனக்குழுவிற்கு உள்ளே வரையறுத்தல் :

இனக்குழுவின் உள்ளே வரையறுக்கப்படும் செயற்கூறானது inline செயற்கூறு போல இயங்குகின்றன. அவை inline உறுப்பு செயற்கூறுகள் எனவும் அழைக்கப்படுகின்றன.

குறிப்பு :

செயற்கூறானது inline செயல்கூறாக அறிவிக்கப்பட்டால் நிரலின் தொகுப்பு நேரத்தில் அச்செயற்கூறு அழைக்கப்படும் இடங்களில் எல்லாம் நிரல் பெயர்ப்பியானது அச்செயற்கூறின் குறிமுறையை பதிலீடு செய்யும்.

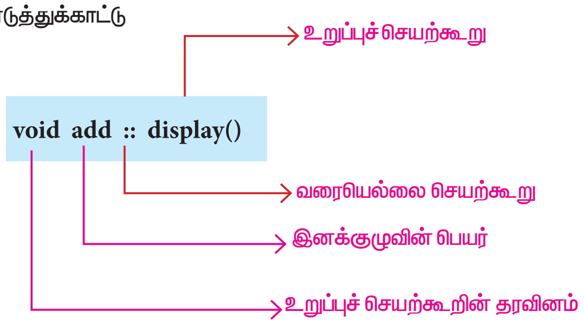
2) இனக்குழுவிற்கு வெளியே வரையறுத்தல் :

இனக்குழுவின் உறுப்பு செயற்கூறானது இனக்குழுவிற்கு வெளியே சாதாரண இனக்குழு வரையறை போலவே வரையறுக்க முடியும். (செயற்கூறு வரையறை பற்றி நீங்கள் நன்கு அறிந்திருந்தால்) இது Outline உறுப்புச் செயற்கூறுகள் (or) non-inline உறுப்புச் செயற்கூறுகள் எனவும் அழைக்கப்படும். உறுப்பு செயற்கூறுகளை இனக்குழுவிற்கு வெளியே வரையறுப்பதற்கு (::) வரையெல்லை செயற்குறியானது (Scope resolution operator) பயன்படுகிறது. உறுப்பு செயற்கூறை இனக்குழுவிற்கு வெளியே வரையறை செய்வதற்கு உதவும் பொது வடிவம்.

Chapter 14.indd 235 4/11/2020 11:49:09 AM

```
return_type class_name :: function_name (parameter list)
{
        செயற்கூறு வரையறை
}
```

எடுத்துக்காட்டு



விளக்க நிரல் 14.1 Inline மற்றும் Outline உறுப்பு செயற்கூறுகள்

```
# include <iostream>
using namespace std;
class Box
{
        double width;           // no access specifier mentioned
public:
        double length;
void printWidth( )              //inline member function definition
        {
                cout<<"\n The width of the box is.."<<width;
        }
void  setWidth( double w );             //prototype of the function
};
void Box :: setWidth(double w)          // outline member function definition
{
width=w;
}
int main( )
{
Box b;  // object for class Box
b.setWidth(10.0);       // Use member function to set the width.
b.printWidth( );                //Use member function to print the width.
return 0;
}
```

அணுகியல்பு குறிப்பிடப்படவில்லை எனில் அவை தானமைவாக Private என கருதப்படும்.

வெளியீடு :

The width of the box is .. 10

குறிப்பு

பல கூற்றுகள், மடக்குகள், switch அல்லது GoTo கூற்றுகள் ஆகியவற்றை கொண்ட உறுப்பு செயற்கூறை Inline செயற்கூறாக அறிவிப்பது ஏற்புடையதல்ல

14.2 பொருள்களை உருவாக்குதல்:

இனக்குழு அறிவிப்பு என்பது இனக்குழுவின் பண்புக்கூறுகளை அறிவிப்பதாகும். இனக்குழுவைப் பயன்பாட்டிற்கு கொண்டுவர அந்த இனக்குழுவில் மாறிகளை அறிவிக்க வேண்டும். இனக்குழுவில் அறிவிக்கப்படும் மாறியானது பொருள்கள் (objects) எனப்படும். பொருள்கள் இனக்குழுவின் சான்றுரு எனவும் அழைக்கப்படும்.

Chapter 14.indd 236 4/11/2020 11:49:09 AM

(எ.கா)

student s;

மேற்கண்ட கூற்றில் s என்பது student இனக்குழுவின் சான்றுருவாகும்.

இனக்குழுவின் பொருளானது இரண்டு முறைகளில் உருவாக்கப்படுகிறது.

1) முழுதளாவிய பொருள் (Global Object)

2) உள்ளமை பொருள் (Local Object)

1) முழுதளாவிய பொருள் (Global Object)

அனைத்து செயற்கூறுகளின் உடற்பகுதிக்கு வெளியில் அல்லது இனக்குழுவின் அறிவிப்பை முடித்து வைக்கும் கொக்கி வடிவ அடைப்பை (braces) தொடர்ந்து பொருளானது அறிவிக்கப்பட்டால் அது முழுதளாவிய மாறி எனப்படும். இனக்குழுவில்/நிரலில் உள்ள அனைத்து செயற்கூறுகளும் அப்பொருளைப் பயன்படுத்திக் கொள்ள முடியும்.

2) உள்ளமை பொருள்கள் (Local Objects)

ஒரு பொருளானது செயற்கூறினுள் அறிவிக்கப்பட்டால் அது உள்ளமை பொருள் எனப்படும். செயல்கூறுக்கு வெளியில் இருந்து இப்பொருளை அணுக முடியாது.

14.2 உள்ளமை பொருளின் பயன்பாட்டை விளக்கும் நிரல்

```
# include <iostream>
# include <conio>
using namespace std
class add
{
  int a,b;
  public:
    int sum;
  void getdata()
  {
    a=5;
    b=10;
    sum = a+b;
  }
} a1;                    //global object
add a2;                  //global object
int main()
{
add a3;          // Local object
a1.getdata();
a2.getdata();
a3.getdata();
cout<<a1.sum;
cout<<a2.sum;
cout<<a3.sum;
return 0;
}
```

முழுதளாவிய பொருளை முழுதளாவிய இனக்குழுவிற்கு மட்டுமே அறிவிக்க முடியும். இனக்குழுவின் வரைறையானது அனைத்து செயற்கூறுகளின் உடற்பகுதிக்கு வெளியே குறிப்பிடப்பட்டால் அது முழுதளாவிய இனக்குழுவாகும்.

வெளியீடு :

```
151515
```


Chapter 14.indd 237 4/11/2020 11:49:09 AM

செயல்பாடு 2

பின்வரும் நிரல் குறிமுறையில் உள்ள பிழைகளை கண்டறிக

```
class A
{
        float x;
        void init()
        {
        A a1;
        X1=1.5;
        }};
void main()
{    A1.init(); }
```

14.3 பொருள்களுக்கான நினைவக இட ஒதுக்கீடு

இனக்குழு வரையறுப்பின் ஒரு பகுதியாக உறுப்பு செயற்கூறுகள் வரையறுக்கப்பட்டால் மட்டுமே அவை உருவாக்கப்பட்டு நினைவகத்தில் இருத்தப்படுகின்றன. அந்த இனக்குழுவைச் சார்ந்த அனைத்துப் பொருள்களும் ஒரே உறுப்பு செயற்கூறை பயன்படுத்திக் கொள்வதால், ஒவ்வொரு பொருள் உருவாக்கப்படும் போதும் உறுப்பு செயற்கூறுகளுக்கு தனித்தனி நினைவக இடம் ஒதுக்கப்படுவதில்லை. உறுப்பு மாறிகளுக்குத் தேவையான நினைவக இடம் மட்டும் ஒவ்வொரு பொருள் உருவாக்கப்படும் போதும் தனித்தனியே ஒதுக்கப்படும். ஏனெனில் ஒவ்வொரு பொருளின் தரவு உறுப்புகள் / உறுப்பு மாறிகள் வெவ்வேறு மதிப்புகளைக் கொண்டிருக்கும்.

14.3 பொருளுக்கான நினைவக இட ஒதுக்கீட்டை விளக்கும் நிரல்

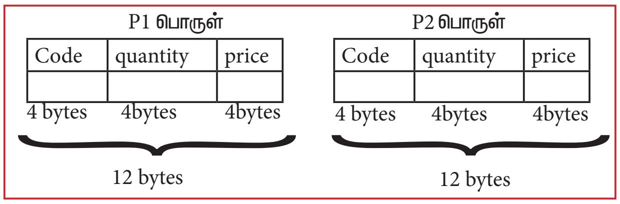
```
# include <iostream>
using namespace std;
class product
{
        int code, quantity;
        float price;
        public:
        void assignData();
        void Print();
};
int main()
{
        product p1, p2;
        cout<<"\n Memory allocation for object p1 " <<sizeof(p1);
        cout<<"\n Memory allocation for object p2 " <<sizeof(p2);
        return 0;
}
```

இனக்குழுவில் பொருளானது உருவாக்கப்பட்ட பின்னர் தான் அதன் உறுப்புகளுக்கு நினைவக இடமானது ஒதுக்கப்படும்.

வெளியீடு :

```
        Memory allocation for object p1  12
        Memory allocation for object p2  12
```

நிரலில் உள்ள assign data () மற்றும் print () உறுப்பு செயற்கூறுகளுக்குப் பொதுவான நினைவகப் பகுதி ஒதுக்கீடு செய்யப்படும். P1 மற்றும் P2 ஆகிய இரண்டு பொருள்களுக்கான தரவு உறுப்புகள் அனைத்தும் பொதுவானப் பகுதியில் உள்ள உறுப்பு செயற்கூறுகளை அணுக முடியும்.



P1 மற்றும் P2 பொருள்களுக்கான நினைவக ஒதுக்கீடு விளக்கம்

Chapter 14.indd 238 4/11/2020 11:49:09 AM

செயல்பாடு 3

கீழ்கண்ட குறிமுறையில் S1 மற்றும் S2 பொருள்களுக்கான நினைவக அளவு யாது?

```
class sum
{
        int n1,n2;
        public:
        void add(){int n3=10;n1=n2=10;}
} s1,s2;
```

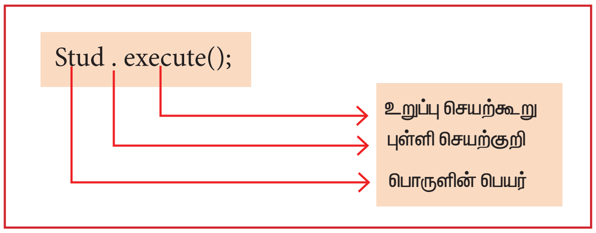
14.4 இனக்குழு உறுப்புகளை அணுகுதல்

இனக்குழுவின் உறுப்புகளை அணுகுவதற்கு புள்ளி (. DOT operator) செயற்குறியானது பயன்படுகிறது. இனக்குழு பொருளின் பெயரைத் தொடர்ந்து புள்ளி செயற்குறியும், அதைத் தொடர்ந்து அணுக வேண்டிய தரவு உறுப்பு / உறுப்பு செயற்கூறின் பெயரைத் தர வேண்டும்.

இனக்குழுவின் உறுப்புகளை அணுகப் பயன்படும் புள்ளி செயற்குறியின் தொடரியல்:

பொருளின் பெயர் . உறுப்பு செயற்கூறின் பெயர் (மெய்யான அளபுருக்கள்) ;

(எடுத்துக்காட்டு)



விளக்க நிரல் 14.4 பொருளைத் தொடர்புகொள்வதை விளக்கும் நிரல்

> பொருளின் உறுப்புச் செயற்கூறை அழைத்தல் என்பது பொருளுக்கு செய்தியை அனுப்புவது அல்லது தொடர்பு கொள்ளுதல் ஆகும்.

```
#include<iostream>
using namespace std;
class compute
{
        int n1,n2;              //private
public :
        int n;
        int add (int a, int b)  //inline member function
        {
                int c=a+b;      //int c ; local variable for this function
                return c;
        }
}c1,c2;
int main()
{
        c1.n =c1.add(12,15);  //member function is called
        c2.n =c2.add(8,4);
        cout<<"\n Sum of object-1  "<<c1.n;
        cout<<"\n Sum of object-2  "<<c2.n;
        cout<<"\n Sum of the two objects are  "<<c1.n+c2.n;
        return 0;
}
```

வெளியீடு :

```
Sum of object-1  27
Sum of object-2  12
Sum of the two objects are  39
```

குறிப்பு

இனக்குழு உறுப்புகளை (பொருள்களை) வரிசையாகக் கொண்ட அணிகள் பொருள்களின் அணிகள் எனப்படும். மற்ற வகையான அணிகளை அறிவித்து வரையறுப்பது போன்றே பொருள்களின் அணிகளையும் உருவாக்கலாம்.

(எ.கா) **student s [10] ;**

10 இனக்குழு பொருள்கள் உருவாக்கப்படுகிரது

Chapter 14.indd 239 4/11/2020 11:49:09 AM

14.5 ஆக்கிகள் – ஓர் அறிமுகம்

இனக்குழுவை வரையறுப்பதன் மூலம் ஒரு புதிய பயனர் வரையறுக்கும் தரவினத்தை உருவாக்க முடியும். இனக்குழுவின் சான்றுருவை சான்றுருவாக்கள் (உருவாக்கி, தொடங்கி வைத்தல்) அவசியமாகும். ஆக்கிகளைப் பயன்படுத்தி பொருள்களைச் சான்றுருவாக்குதல் முடியும்.

14.5.1 ஆக்கிகளின் தேவை

C++ ல் உள்ள அணி அல்லது கட்டுரு, அதன் அறிவிக்கும் நேரத்தில் தொடங்கி வைக்கப்படுகிறது.

எடுத்துக்காட்டாக

```
struct sum
  {
        int n1,n2;
  };
class add
  {
        int num1,num2;
  };
int  main()
  {
        int arr[]={1,2,3};       //declaration and initialization of array
        sum s1={1,1};           //declaration and initialization of structure object
        add  a1={0,0};          // class object declaration and initialization throws
                                                       compilation error
   }
```

இனக்குழு உறுப்பு செயற்கூறானது அதனோடு தொடர்புடைய அணுகியல்பு எதுவாக இருந்தாலும் அணுக முடியும்.

இனக்குழு பொருள் அதன் அறிவிப்பு நேரத்தில் தொடங்கி வைப்பது (initialization) கட்டுரு அல்லது அணியை தொடங்குதல் போன்றது அல்ல. ஏனென்றால், இனக்குழு உறுப்புகள் அதனோடு தொடர்புடைய அணுகியல்பைக் கொண்டிருக்கும் (public, private or protected). அதனால், இனக்குழுக்கள் ஆக்கிகள் என்னும் சிறப்பு உறுப்பு செயற்கூறுகளைக் கொண்டிருக்கும். ஆக்கி செயற்கூறு இனக்குழு பொருளைத் தொடங்கி வைக்கிறது.

14.6 அறிவிப்பு மற்றும் வரையறுப்பு

ஓர் இனக்குழுவின் சான்றுரு பயன்பாட்டுக்கு வரும்போது ஆக்கி எனப்படும் சிறப்பு செயற்கூறு இயக்கப்படுகிறது. ஆக்கியின் பெயர் இனக்குழுவின் பெயராகவே இருக்க வேண்டும். இது எதையும் திருப்பி அனுப்பாது. இதற்கு எந்த தரவினத்தோடும் தொடர்பு இல்லை.

இனக்குழு வரையறுப்பின் உள்ளே அல்லது வெளியே இதனை வரையறுக்கலாம்.

எடுத்துக்காட்டு : 1

பின்வரும் நிரல் 14.5 ஆக்கியை இனக்குழுவின் உள்ளே வரையறுப்பதை விளக்குகிறது.

```
#include<iostream>
using namespace std;
class Sample
      {
int i,j;
            public :
int k;
              Sample()
              {
                 i=j=k=0;//constructor defined inside the class
              }
      };
```

14.6.1 ஆக்கியின் செயல்பாடுகள் :

இனக்குழுவின் சான்றுரு உருவாக்கப்படும் போது அல்லது அறிவிப்பின் போது, சிறப்பு தொடங்கு உறுப்பு செயற்கூறான ஆக்கி தானாகவே தொடங்கி இயக்கப்படுகிறது. ஆக்கியின் முக்கிய செயல்பாடுகள் ஆவன :

1. பொருளுக்கு நினைவகத்தில் இடம் ஒதுக்குகிறது

2. இனக்குழுப் பொருளின் உறுப்புகளில் தொடக்க மதிப்பு இருத்துகிறது.

இனக்குழுப் பொருளைத் தொடங்கி வைப்பதற்கான மாற்றுவழி உள்ளது. ஆனால் இந்த வழியில் உறுப்பு செயற்கூறினை வெளியில் இருந்து அழைக்க வேண்டும்.

14.7 ஆக்கிகளின் வகைகள்

தானமைவு ஆக்கிகள் :-

அளபுருக்களை ஏற்காத ஆக்கி தானமைவு ஆக்கி என்றழைக்கப்படுகிறது. எடுத்துக்காட்டாக Data என்கிற இனக்குழுவில் Data::Data() என்பது தானமைவு ஆக்கி. இந்த ஆக்கியை பயன்படுத்தி பொருள்களை உருவாக்குவது, பிற தரவு வகைகளைக் கொண்ட மாறிகளை உருவாக்குவது போன்றதாகும்.இனக்குழுவில் வெளிப்படையாக ஆக்கிகள் இல்லாத நிலையில் (பயனர் வரையறுக்கும் ஆக்கி) நிரல் பெயர்ப்பி தானாகவே தானமைவு ஆக்கியை உள்ளார்ந்த (implicitly) inline public உறுப்பாக சேர்த்துக் கொள்ளும்.

அளபுரு ஏற்கும் ஆக்கி (Parameterized)

அளபுருக்களை ஏற்கும் ஆக்கி, அளபுரு ஏற்கும் ஆக்கியாகும். வேறுபட்ட தொடக்க மதிப்புகளைக் கொண்ட பொருள்களை உருவாக்க இந்த வகை ஆக்கி பயன்படுகிறது. செயற்கூறுக்கு அளபுருக்களை அனுப்புவதன் மூலம் இது சாத்தியமாகிறது.

எடுத்துக்காட்டு:

Data::Data(int,int)

நகல் ஆக்கிகள் :

ஏற்கெனவே உள்ள இனக்குழுவின் பொருளுக்குக் குறிப்புகளைக் கொண்ட ஆக்கிக்கு நகல் ஆக்கி என்று பெயர். மாறாக, நகல் ஆக்கி என்பது ஆக்கியின் ஒரு வகை, இது ஏற்கெனவே உள்ள இனக்குழுவின் பொருளை நகலெடுக்கப் பயன்படுகிறது. இதன் பொதுவான அமைப்பு இவ்வாறு இருக்கும் **Data (Data &);** இதில் **Data** என்பது இனக்குழுப் பெயர்

நகல் ஆக்கி கீழ்காணும் சூழல்களில் இயக்கப்படும்.

1) ஏதேனும் ஓர் உறுப்புச் செயற்கூறினுக்கு ஒரு பொருளை அளபுருவாக அனுப்பி வைக்கும் போது.

   எடுத்துக்காட்டு : void Data :: putdata (Data x);

2) ஓர் உறுப்புச் செயற்கூறு ஒரு பொருளைத் திருப்பியனுப்பும் போது.

   எடுத்துக்காட்டு : Data getdata ()

3) ஒரு பொருளைக் குறிப்புவகையாக இனக்குழுவின் சான்றுருக்கு அனுப்பி வைக்கப்படும்போது. எடுத்துக்காட்டு : Data D1, D2(D1) //D2(D1) நகல் ஆக்கியை அழைக்கிறது.

பின்வரும் நிரல் **14.6** ஆக்கிகளின் வகைகள்

```
#include<iostream>
using namespace std;
class Data
 {
     int i, j;
   public:
     int k;
      Data()
       {
          cout<<"Non Parametrerized constructor";
           i=0;
            j=0'
         }
       Data(int a,int b)
```

```
    {
        cout<<"Parametrerized constructor";
         i=a;
          j=b'
    }
   Data(Data &a)
    {
        cout<<"Copy constructor";
         i=a.i;
          j=b.j'
    }

      void display()             //member function
       {
         cout<< i<<j;
        }
   };
int main()
{
      Data  d1,d2(10,20),d3(d2);
      d1.display();
      d2.display();
      d3.display();
       return 0;

}
```

தானமைவு ஆக்கியின் தனிச்சிறப்பு :

தானமைவு ஆக்கிகள், குறிப்பிட்ட தொடக்க மதிப்பு இல்லாத பொருள்களை உருவாக்கப் பயன்படுகிறது. பொருள்களின் அணியை உருவாக்கவும் இது பயன்படுகிறது.

14.8 ஆக்கிகளை அழைத்தல்

இரண்டு வழிகளில் அளபுருக்களை ஏற்கும் ஆக்கியைப் பயன்படுத்திப் பொருளை உருவாக்கலாம்.

1. உள்ளார்ந்த அழைப்பு (Implicit call)
2. வெளிப்படையான அழைப்பு (Explicit call)

14.8.1 உள்ளார்ந்த அழைப்பு:- (Implicit call)

இந்த முறையில், எப்பொழுதெல்லாம் பொருள் உருவாக்கப்படுகிறதோ அப்பொழுது அளபுருக்களை ஏற்கும் ஆக்கி தானாகவே அழைக்கப்படுகிறது. எடுத்துக்காட்டாக simple s1 (10, 20); இதில் s1 என்ற பொருளை உருவாக்கும்போது அளபுருவை ஏற்கும் ஆக்கி தானாகவே அழைக்கப்படுகிறது.

14.8.2 வெளிப்படையான அழைப்பு :- (Explicit call)

இந்த முறையில் ஆக்கியின் பெயரை வெளிப்படையாக கொடுக்கும்போது அளபுருக்களை ஏற்கும் ஆக்கி அழைக்கப்படுகிறது. அதனால் பொருளை உருவாக்கி தொடங்கி வைக்க முடிகிறது.

எடுத்துக்காட்டாக,

```
simple s1=simple(10,20);      //வெளிப்படையான அழைப்பு
```

வெளிப்படையான அழைப்பு முறை மிகவும் பொருத்தமானது. ஏனென்றால் இதில் தற்காலிக பொருளை உருவாக்குவதால் தரவு இழப்புக்கான வாய்ப்பு ஏற்படாது. தற்காலிக பொருள் கோவையில் பயன்படுத்தும் வரை நினைவகத்தில் இருக்கும். அதன்பிறகு அது அழிந்து விடும்.

Chapter 14.indd 242 4/11/2020 11:49:10 AM

14.9 பொருள் இயங்குநிலையில் தொடங்குதல்

இயங்கு நேரத்தில் தொடக்க மதிப்புகள் கொடுக்கப்பட்டால் அது இயங்கு நிலையில் தொடங்குதல் எனப்படும்.

நிரல் 14.7 இயங்குநிலை தொடங்கலை எடுத்துக்காட்டுகிறது

```
#include<iostream>
using namespace std;
class X
  {
int n;
    float avg;
    public:
          X(int p,float q)
          {  n=p;
          avg=q;
                      }
void disp()
{
          cout<<"\n Roll numbe:- " <<n;
          cout<<"\nAverage :- "<<avg;
}
    };
int main()
{
int a ; float b;
          cout<<"\nEnter the Roll Number";
          cin>>a;
          cout<<"\nEnter the Average";
          cin>>b;
          X x(a,b);    // dynamic initialization
          x.disp();
           return 0;
}
```

வெளியீடு :

```
          Enter the Roll Number 1201
          Enter the Average 98.6
          Roll numbe:- 1201
          Average :- 98.6
```

14.10 ஆக்கியின் தனிச்சிறப்புப் பண்புகள்

- ஆக்கியின் பெயர் இனக்குழுவின் பெயராகவே இருக்க வேண்டும்.
- ஆக்கி எந்த மதிப்பையும் திருப்பி அனுப்பாது.
- ஆக்கி அளபுருக்களின் பட்டியலைக் கொண்டிருக்கும்.
- ஆக்கி செயற்கூறு, பணி மிகுக்கப்பட முடியும்.
- ஆக்கியைத் தருவிக்க முடியாது. ஆனால் தருவிக்கப்பட்ட இனக்குழு அடிப்படை இனக்குழுவின் ஆக்கியை அழைக்க முடியும்.
- பயனர் வரையறுக்கும் ஆக்கி இல்லாத போது நிரல்பெயர்ப்பி ஓர் ஆக்கியை உருவாக்கிக் கொள்ளும்.
- நிரல்பெயர்ப்பி உருவாக்கும் ஆக்கி Public உறுப்பு செயற்கூறாகும்.
- ஓர் இனக்குழு பொருள் உருவாக்கப்படும்போது ஆக்கி தானாகவே இயக்கப்படும்.
- ஒரு புதிய பொருளை அதன் இனக்குழுவில் உருவாக்க ஆக்கியை வெளிப்படையாகப் பயன்படுத்த முடியும்.

14.11 அழிப்பிகள்

அழிப்பி என்பது, ஒரு பொருளை உருவாக்கும்போது, ஆக்கியால் பொருளுக்கென ஒதுக்கப்படும் நினைவகப் பகுதியை விடுவிக்கும் ஒரு செயற்கூறாகும். இதுவும் இனக்குழுவின் பெயரையேக் கொண்டிருக்கும். ஆனால் ~ என்னும் குறியைப் பெயரின் முன்னொட்டாக கொண்டிருக்கும். இது எந்த மதிப்பையும் திருப்பி அனுப்புவதில்லை, எந்த தரவினத்தோடும் தொடர்புடையவை அல்ல.

14.11.1 அழிப்பியின் தேவை

அழிப்பியின் நோக்கம் ஒரு பொருள் அதன் வாழ்நாளில் பெற்ற வளங்களை விடுவிப்பதாகும். ஒரு பொருளை உருவாக்கும் போது ஆக்கியால் பொருளுக்கென ஒதுக்கப்பட்ட நினைவகப் பகுதியை அழிக்கும்.

14.11.2 அறிவிப்பு மற்றும் வரையறுப்பு

அழிப்பி என்ற சிறப்பு செயற்கூறானது ஆக்கியால் உருவாக்கப்பட்ட பொருளின் வாழ்நாள் முடிந்து அழியும்போது அழைக்கப்படுகிறது. பொதுவாக Public பகுதியில் அறிவிக்கப்படுகிறது.

Chapter 14.indd 243 4/11/2020 11:49:10 AM

நிரல் 14.8 அழிப்பி செயற்கூறினை விளக்குகிறது

```
#include<iostream>
using namespace std;
class simple
{
private:
int a, b;
public:
simple()
{
a= 0 ;
b= 0;
cout<< "\n Constructor of class-simple ";
}
void getdata()
{
cout<<"\n Enter values for a and b ";
cin>>a>>b;
}
void putdata()
{
cout<<"\nThe two integers are .. ";
cout<<<<a<<'\t'<< b<<endl;
cout<<"\n The sum = "<<a+b;
}
~simple()
{    cout<<"\n Destructor is executed ";}
};
int main()
{
simple s;
s.getdata();
s.putdata();
return 0;
}
```

Output:

```
Constructor of class-simple
Enter values for a and b  6 7
The two integers are .. 6      7
The sum = 13
Destructor is executed
```

14.12 அழிப்பியின் தனிச் சிறப்புப் பண்புகள்

1) அழிப்பியின் பெயரானது ~ என்ற முன்னொட்டு குறியுடன் கூடிய இனக்குழுவின் பெயரையேக் கொண்டிருக்கும்.

2) அழிப்பி, செயலுருபுகளை ஏற்காது.

3) அழிப்பி எந்த மதிப்பையும் திருப்பி அனுப்பாது.

4) அழிப்பி பணிமிகுக்கப்பட முடியாது. அதாவது ஓர் இனக்குழுவில் ஓர் அழிப்பி மட்டுமே இருக்க முடியும்.

5) பயனர் அழிப்பியை வரையறுக்காத போது நிரல்பெயர்ப்பி ஓர் அழிப்பியை உருவாக்கிக் கொள்ளும்.

6) நிரலில் உருவாக்கப்பட்ட ஓர் இனக்குழு பொருளின் பயன்பாடு முடிவுக்கு வரும்போது அழிப்பி தானாகவே இயக்கப்படும்.

7) இதை தருவிக்க (மரபுவழி) முடியாது.

நினைவில்கொள்க

- தரவையும் அதனோடு தொடர்புடைய செயற்கூறினையும் இணைத்து வைப்பது இனக்குழுவாகும்.
- C++ -ல் உள்ள பயனர் வரையறுக்கும் தரவினமான இனக்குழுவைப் பயன்படுத்திப் பொருளை உருவாக்கலாம்.
- ஓர் இனக்குழுவை அறிவிக்கும் பொழுது தரவு உறுப்புகள், உறுப்பு செயற்கூறுகள், அணுகியல்பு வரையறுப்பிகள் மற்றும் இனக்குழுவின் பெயர் கொடுக்கப்படுகிறது.
- ஓர் இனக்குழுவின் உறுப்பு செயற்கூறினை அந்த இனக்குழுவின் உள்ளே அல்லது வெளியே வரையறுக்கலாம்
- இனக்குழுவின் public உறுப்புகளை அந்த இனக்குழு பொருளைப் பயன்படுத்தி இனக்குழுவின் வெளியிலிருந்து நேரடியாக அணுகலாம்.
- ஓர் இனக்குழுவின் OOP – சிறப்பம்சமான உறைப்பொதியாக்கத்தைத் தரவு மற்றும் தொடர்புடைய செயற்கூறுகளை ஒன்றாக சேர்த்து வைப்பதன் மூலம் ஆதரிக்கிறது.
- Private மற்றும் protected மூலம் வெளி உலகிற்கு தகவலை மறைப்பதன் மூலம் தரவு மறைத்தலை இனக்குழு ஆதரிக்கிறது.

நினைவில்கொள்க

- ஒரு இனக்குழுவின் உறுப்பு செயற்கூறினை மற்றொரு உறுப்பு செயற்கூறு மூலம் அழைத்தல் பின்னலான உறுப்பு செயற்கூறு என்றழைக்கப்படுகிறது.
- வரையெல்லை உணர்த்தும் செயற்குறி ::, ஒரு இனக்குழு பெயரில் பயன்படுத்தும் class-name :: function name பயன்படுத்தும் போது அந்த இனக்குழுவின் உறுப்புகளைக் குறிக்கிறது. :: variable-name, முழுதளாவிய மாறியைக் குறிக்கிறது.
- ஓர் இனக்குழுவின் சான்றுரு பயன்பாட்டுக்கு வரும்போது ஆக்கி எனப்படும் சிறப்பு செயற்கூறு இயக்கப்படுகிறது.
- ஆக்கி செயற்கூறு இனக்குழு பொருளுக்கு நினைவகத்தில் இடம் ஒதுக்குகிறது மற்றும் தொடங்கி வைக்கிறது.
- அழிப்பி என்ற சிறப்பு செயற்கூறானது ஆக்கியால் உருவாக்கப்பட்ட பொருளின் வாழ்நாள் முடிந்து அழியும்போது அழைக்கப்படுகிறது.
- ஆக்கிகள் மற்றும் அழிப்பிகள் பெயர் இனக்குழுவின் பெயராக இருக்க வேண்டும்.
- அளபுருக்கள் இல்லாத ஆக்கி தானமைவு ஆக்கி என்றழைக்கப்படுகிறது.
- தானமைவு செயலுருபுகளைக் கொண்ட ஆக்கி தானமைவு ஆக்கிக்கு சமமானதாகும்.
- ஆக்கி மற்றும் அழிப்பி எதையும் திருப்பி அனுப்பாது. எந்த தரவினத்தோடும் தொடர்புடையது அல்ல.
- பொருளை இயங்குநிலையில் தொடங்க முடியும்.

பயிற்சி

1 பின்வரும் குறிப்புகளைப் பயன்படுத்தி Employee இனக்குழுவை வரையறுக்கவும்

Employee-இனக்குழுவின் private உறுப்புகள்

empno- integer
ename – 20 characters
basic – float
netpay, hra, da, - float
calculate () – basic+hra+da -வைக் கண்டுப்பிடித்து float மதிப்பைத் திருப்பி அனுப்பும் செயற்கூறு

employee - இனக்குழுவின் public செயற்கூறுகள்

havedata() – (empno, ename, basic, hra, da) உள்ளீடாகப்பெற்று netpay கணக்கிட calculate() செயற்கூறை அழைக்கும் ஒரு செயற்கூறு

dispdata() – அனைத்து தரவு உறுப்புகளையும் திரையில் வெளியிடுவதற்கான செயற்கூறு

மதிப்பீடு

பகுதி – அ

1. ஓர் இனக்குழுவுக்குள் அறிவிக்கப்படும் மாறிகளை தரவு உறுப்புகள் என குறிப்பிடுகின்றோம். செயல்கூறுகளை எவ்வாறு குறிப்பிடுகிறோம்.

   (அ) தரவு செயற்கூறுகள்
   (ஆ) inline செயற்கூறுகள்
   (இ) உறுப்பு செயற்கூறுகள்
   (ஈ) பண்புக் கூறுகள்

2. பின்வரும் உறுப்புச் செயற்கூறினைப் பற்றிய கூற்றுகளில் எது சரி அல்லது தவறு ?

   i) புள்ளி செயற்குறி மூலம் ஒரு உறுப்புச் செயற்கூறு, இன்னொரு உறுப்புச் செயற்கூறினை நேரடியாக அழைக்கலாம்.
   ii) இனக்குழுவின் private தரவுகளை உறுப்புச் செயற்கூறு அணுக முடியும்.

   (அ) i - சரி, ii - சரி
   (ஆ) i – தவறு, ii - சரி
   (இ) i -தவறு, ii - சரி
   (ஈ)i – தவறு, ii – தவறு

Chapter 14.indd 245 4/11/2020 11:49:10 AM

3. ஒரு உறுப்பு செயற்கூறு, இன்னொரு உறுப்பு செயற்கூறைப் புள்ளி செயற்குறியைப் பயன்படுத்தாமல் நேரடியாக அணுகலாம் என்பதை எவ்வாறு குறிப்பிடலாம்.

(அ) துணை செயற்கூறு

(ஆ) துணை உறுப்பு

(இ) பின்னலான உறுப்பு செயற்கூறு

(ஈ) துணை உறுப்பு செயற்கூறு

4. இனக்குழுவுக்குள் வரையறுக்கப்படும் செயற்கூறுகள் எந்த செயற்கூறுகளைப் போல் இயங்குகின்றன?

(அ) inline செயற்கூறுகள்

(ஆ) inline அல்லாத செயற்கூறுகள்

(இ) Outline செயற்கூறுகள்

(ஈ) தரவு செயற்கூறு

5. பின்வரும் எந்த அணுகியல்பு வரையறுப்பி தவறுதலான மாற்றங்களிலிருந்து தரவைப் பாதுகாக்கிறது?

(அ) Private (ஆ) Protected
(இ) Public (ஈ) முழுதளாவிய

6. கீழ்கண்ட நிரலில் எத்தனை பொருள்கள் உருவாக்கப்பட்டுள்ளன ?

```
class x
{
int y;
public:
x (int z) {y=z;}
} x1 [4];
int main ()
{ x x2(10);
return 0;}
```

(அ) 10 (ஆ) 14
(இ) 5 (ஈ) 2

7. ஆக்கி செயற்கூறு பற்றிய பின்வரும் கூற்றுகள் சரியா, தவறா எனக் கூறு.

i) ஆக்கிகள் private பகுதியில் அறிவிக்கப்பட வேண்டும்

ii) பொருள்கள் உருவாக்கப்படும் போது, ஆக்கி தானாகவே இயக்கப்படும்.

(அ) சரி, சரி (ஆ) சரி, தவறு

(இ) தவறு, சரி (ஈ) தவறு தவறு

8. பின்வரும் முன்வடிவுக்கு கீழ்கண்டவற்றுள் எந்த ஆக்கி இயக்கப்படும் ?

add display (add &); //add என்பது இனக்குழுவின் பெயர்

(அ) தானமைவு ஆக்கி

(ஆ) அளபுருக்களுடன் கூடிய ஆக்கி

(இ) நகல் ஆக்கி

(ஈ) அளபுருக்கள் இல்லாத ஆக்கி

பகுதி – ஆ

குறு வினாக்கள்

1. உறுப்புகள் என்றால் என்ன?

2. பயனர் வரையறுத்த தரவினம் வகையான கட்டுரு, இனக்குழு – வேறுபடுத்திக் காட்டுக.

3. பொருள் நோக்கு நிரலாக்கு குறிமுறை **(OOP)** அடிப்படையில் இனக்குழு மற்றும் பொருள் பற்றி வேறுபடுத்திக் காட்டுக.

4. நிரல்பெயர்ப்பி தாமாகவே ஆக்கியை உருவாக்கிக் கொள்ள முடிந்தாலும், ஆக்கி வரையறுப்பு ஏன் சிறந்த வழக்கம் என்று கருதப்படுகிறது ?

5. அழிப்பியின் முக்கியத்துவத்தைப் பற்றி எழுதுக.

பகுதி – இ

சிறு வினாக்கள்

1. பின்வரும் நிரலில் கட்டளை அமைப்புப் பிழை ஏதேனும் இருப்பின், அவற்றை நீக்கி, பிழையைக் கோடிட்டு காட்டி, நிரலை மாற்றி எழுதவும்.

```
#include<iostream>
#include<stdio.h>
  class mystud
  { int studid =1001;
  char name[20];
public
  mystud( )
  { }
void register ( ) {cin>>stdid;gets(name);
}
void display ( )
  { cout<<studid<<": "<<name<<endl;}
}
int main( )
  { mystud MS;
  register.MS( );
  MS.display( );
}
```

2. நிரலின் இயங்கு நேரத்தில் ஒரு பொருளை எவ்வாறு தொடங்கி வைப்பது என்பதை ஒரு எடுத்துக்காட்டுடன் எழுது.

3. Public அணுகுமுறையில் ஆக்கிகள், அழிப்பிகள் அறிவிப்பினால் விளையும் நன்மைகள் யாவை?

4. கீழே கொடுக்கப்பட்டுள்ள C++ நிரலைக் கொண்டு (i) & (ii) வினாக்களுக்கான விடைகளைத் தருக.

```
class TestMeOut
{
public:
  ~TestMeOut() //Function 1
  {cout<<"Leaving the examination hall"<<endl;}
  TestMeOut() //Function 2
  {cout<<"Appearing for examination"<<endl;}
void MyWork() //Function 3
  {cout<<"Attempting Questions//<<endl;}
};
```

(i) பொருள்நோக்கு நிரலாக்க முறையின்படி, செயற்கூறு-1 என்பது எதைக் குறிக்கிறது, எப்பொழுது அது அழைக்க / இயக்கப்படுகிறது ?

(ii) பொருள்நோக்கு நிரலாக்க முறையின்படி, செயற்கூறு-2 என்பது எதைக் குறிக்கிறது, எப்பொழுது அது இயக்க / அழைக்கப்படுகிறது ?

பகுதி – ஈ

பெருவினாக்கள்

1. ஆக்கி, அழிப்பி – வேறுபாடு தருக

2. கீழ்காணும் வரையறுப்புகளுடன் Resort என்னும் ஓர் இனக்குழுவை வரையறுக்கவும்

private உறுப்புகள்

Rno // அறை எண்ணை இருத்தி வைக்கும் தரவு உறுப்பு
Name //பயனரின் பெயரை இருத்தி வைக்கும் தரவு உறுப்பு
Charges // ஒரு நாளுக்குரிய கட்டணத்தை இருத்தி வைக்கும் தரவு உறுப்பு
Days // நாட்களின் எண்ணிக்கையை இருத்தி வைக்கும் தரவு உறுப்பு
Compute ( ) // Days * Charges கொண்டு மொத்த தொகையை கணக்கிடும் செயற்கூறு
// மொத்த தொகை 11000 ரூபாய்க்கு மேல் இருந்தால், மொத்த தொகையைக் கணக்கிட 1.02 * Days *Charges
Public member:
getinfo( ) // பெயர், அறைஎண், கட்டணம், நாட்கள் போன்ற தகவல்களை உள்ளீடாகப் பெறும் செயற்கூறு
dispinfo ( ) // உள்ளிடப்பட்ட தரவுகள் மற்றும் Compute செயற்கூறினைப் பயன்படுத்தி கணக்கிட மொத்த தொகையை வெளியிடும் செயற்கூறு

3. கீழ்காணும் நிரலுக்கு வெளியீடு எழுது.

```
#include<iostream>
using namespace std;
class student
{
        int rno, marks;
        public:
        student(int r,int m)
        { cout<<"Constructor "<<endl;
          rno=r;
          marks=m;
        }
        void printdet()
        {
          marks=marks+30;
          cout<<"Name: Bharathi"<<endl;
          cout<<"Roll no : "<<rno<<"\n";
          cout<<"Marks : "<<marks<<endl;
          }
};
int main()
{
        student s(14,70);
        s.printdet();
        cout<< "Back to Main";
        return 0;
}
```

துணை நூல்கள்:

(1) Object Oriented Programming with C++ (4th Edition), Dr. E. Balagurusamy, Mc.Graw Hills.

(2) The Complete Reference C++ (Forth Edition), Herbert Schildt.Mc.Graw Hills.

Chapter 14.indd 247 4/11/2020 11:49:10 AM