

جامعة المنارة

مقرر برمجة 2

د. هلا نصار

م. أوشين داود

جامعة
المنارة

HANARA UNIVERSITY

محاضرات الأسبوع 5
الفصل الأول - 2022/2021

Classes

أنشئ class اسمه Room واشتق منه object اسمه room1 يحتوي على ثلاث خصائص: الطول - العرض - الارتفاع وتابعين لحساب الحجم والمساحة

```
#include <iostream>
using namespace std;
class Room
{public:
    double length;
    double breadth;
    double height;
    double calculateArea();
    double calculateVolume() ;
};
double Room :: calculateArea()
{return length * breadth;}
double Room :: calculateVolume()
{return length * breadth * height;}
```

```
int main() {
// create object of Room class
Room room1;
// assign values to data members
room1.length = 42.5;
room1.breadth = 30.8;
room1.height = 19.2;
// calculate and display the area and volume of the room
cout << "Area of Room = " << room1.calculateArea() << endl;
cout << "Volume of Room = " << room1.calculateVolume() << endl;
return 0;
}
```

Classes : a pointer to an object - 1

```
#include <iostream>
using namespace std;
class Room
{public:
    double length;
    double breadth;
    double height;
    double calculateArea();
    double calculateVolume() ;
};
double Room :: calculateArea()
{return length * breadth;}
double Room :: calculateVolume()
{return length * breadth * height;}
```

```
int main() {
// create object of Room class
Room room1;
Room *ptr;
ptr=&room1;
// assign values to data members
(*ptr).length = 42.5;
(*ptr).breadth = 30.8;
(*ptr).height = 19.2;
// calculate and display the area and volume of the room
cout << "Area of Room = " << (*ptr).calculateArea() << endl;
cout << "Volume of Room = " << (*ptr).calculateVolume() << endl;
return 0;
}
```

Classes : a pointer to an object -2

```
#include <iostream>
using namespace std;
class Room
{public:
    double length;
    double breadth;
    double height;
    double calculateArea();
    double calculateVolume() ;
};
double Room :: calculateArea()
{return length * breadth;}
double Room :: calculateVolume()
{return length * breadth * height;}
```

```
int main() {
// create object of Room class
Room room1;
Room *ptr;
ptr=&room1;
// assign values to data members
ptr->length = 42.5;
ptr->breadth = 30.8;
ptr->height = 19.2;
// calculate and display the area and volume of the room
cout << "Area of Room = " << ptr->calculateArea() << endl;
cout << "Volume of Room = " << ptr->calculateVolume() << endl;
return 0;
}
```

Classes : access specifiers – 3

```
#include <iostream>
using namespace std;
class Room {
private:
    double length;
    double breadth;
    double height;
public:
    void initData(double len, double brth, double hgt)
    {length = len;
    breadth = brth;
    height = hgt;}
    double calculateArea()
    {return length * breadth;}
    double calculateVolume()
    {return length * breadth * height;}
};
```

يوجد محددًا وصول أساسيان Public، Private يمكن الوصول إلى خصائص الصنف الموجودة في قسم private فقط عن طريق التتابع الموجودة في قسم public

```
int main() {
// create object of Room class
Room room1;
// pass the values of private variables as arguments
room1.initData(42.5, 30.8, 19.2);
cout << "Area of Room = " << room1.calculateArea() << endl;
cout << "Volume of Room = " << room1.calculateVolume() <<
endl;
return 0;
}
```

Classes : access specifiers – 3

```
#include <iostream>
using namespace std;
class Room {
private:
    double length;
    double breadth;
    double height;
public:
    void initData(double len, double brth, double hgt)
    {length = len;
    breadth = brth;
    height = hgt;}
    double calculateArea()
    {return length * breadth;}
    double calculateVolume()
    {return length * breadth * height;}
};
```

يوجد محددًا وصول أساسيان Public، Private يمكن الوصول إلى خصائص الصنف الموجودة في قسم private فقط عن طريق التوابع الموجودة في قسم public

```
int main() {
// create object of Room class
Room room1;
// pass the values of private variables as arguments
Double x,y,z;
cin>>x>>y>>z;
room1.initData(x,y,z);
cout << "Area of Room = " << room1.calculateArea() << endl;
cout << "Volume of Room = " << room1.calculateVolume() << endl;
return 0;
}
```

main.cpp [dfgnmoi] - Code::Blocks 17.12

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main() : int

Management

Projects Symbols Files

Workspace

dfgnmoi

Sources

main.cpp

```
19 double calculateVolume() {
20     return length * breadth * height;
21 }
22 };
23 int main() {
24
25     // create object of Room class
26     Room room1;
27
28     // pass the values of private variables as arguments
29     room1.initData(42.5, 30.8, 19.2);
30     room1.length = 100;
31     cout << "Area of Room = " << room1.calculateArea() << endl;
32     cout << "Volume of Room = " << room1.calculateVolume() << endl;
33
34     return 0;
35 }
```

It is private!

Logs & others

Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugge

File	Line	Message
		=== Build: Debug in dfgnmoi (compiler: GNU GCC Compiler) ===
C:\Users\HP\Do...		In function 'int main()':
C:\Users\HP\Do...	5	error: 'double Room::length' is private
C:\Users\HP\Do...	30	error: within this context
		=== Build failed: 2 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===

C:\Users\HP\Documents\codeblocks\dfgnmoi\main.cpp C/C++ Windows (CR+LF) WINDOWS-1256 Line 30, Col 1, Pos 604 Insert Read/Write default

2021/11/19 ENG

Classes : objects types (Global / static/local)

```
#include <iostream>
using namespace std;
class Room {
private:
    double length;
    double breadth;
    double height;
public:
    void initData(double len, double brth, double hgt)
    {length = len;
    breadth = brth;
    height = hgt;}
    double calculateArea()
    {return length * breadth;}
    double calculateVolume()
    {return length * breadth * height;}
};
```

```
Room room_global;
int main() {
    static Room room_static;
    // create object of Room class
    Room room1;
    // pass the values of private variables as arguments
    room1.initData(42.5, 30.8, 19.2);
    cout << "Area of Room = " << room1.calculateArea() << endl;
    cout << "Volume of Room = " << room1.calculateVolume() <<
    endl;
    return 0;
}
```


Classes : constructor function - 1

```
// C++ program to demonstrate the use of default constructor
#include <iostream>
using namespace std;
// declare a class
class Wall {
private:
    double length;
public:
// default constructor to initialize variable
    void printing () {cout << length;}
};
int main()
{Wall wall1;
wall1.printing();
return 0;
}
```

Classes : constructor function - 1

```
// C++ program to demonstrate the use of default constructor
#include <iostream>
using namespace std;
// declare a class
class Wall
{private:
    double length;
public:
// default constructor to initialize variable
    Wall()
    {length = 5.5;
    cout << "Creating a wall." << endl;
    cout << "Length = " << length << endl; }
};
int main() {
Wall wall1;
return 0;
}
```

التابع الباني هو تابع عضو في الصنف
يستدعي اوتوماتيكياً عند اشتقاق الغرض
لتخصيص الذاكرة اللازمة
وهو يحمل نفس اسم الصنف ويكون
افتراضياً دون قيمة معادة أو بارامترات
ممررة إليه.
يعمل التابع الباني على إعطاء قيم ابتدائية
لخصائص الصنف.

Classes : constructor function - 2

```
// example: class constructor
#include <iostream>
using namespace std;
class Rectangle
{int width, height;
public:
    Rectangle (int,int);
    int area () {return (width*height);}
};
Rectangle::Rectangle (int a, int b)
{width = a;
 height = b;}
int main ()
{ Rectangle rect1 (3,4);
  Rectangle rect2 (5,6);
  cout << "rect1 area: " << rect1.area() << endl;
  cout << "rect2 area: " << rect2.area() << endl;
  return 0;
}
```

- إمكانية الاستغناء عن الكلمة المفتاحية private فتكون الخصائص افتراضياً من النوع private
- تعريف التابع الباني خارج الصنف وتمييزه بارامترات اليه وهو ما يسمى parametrized constructor

Classes : destructor function - 1

```
// C++ program to demonstrate the use of default constructor
#include <iostream>
using namespace std;
// declare a class
class Wall {
private:
    double length;
public:
// default constructor to initialize variable
    Wall()
    { length = 5.5;    cout << length<<endl; }
    ~Wall () {cout << "the destructor is running now " << endl;}
};
Wall wall1;
int main()
{Wall wall2;
    static Wall wall3;
    return 0;
}
```

التابع الهادم تابع عضو ضمن الصنف مهمته تحرير الذاكرة المشغولة من قبل الغرض يكون له نفس اسم الصنف مسبقاً بـ ~ وليس له لا قيمة معادة ولا بارامترات ويتم استدعاؤه تلقائياً بمجرد الخروج من مجال الرؤية الذي جرى تعريف الغرض ضمنه.

لاحظ في المثال المجاور ترتيب البناء والهدم

حسب مجالات الرؤية للأغراض المختلفة

البناء يتم وفق ترتيب الاشتقاق
الهدم يتم بترتيب معاكس حيث تهدم أولاً
الأغراض المحلية ثم الستاتيكية ثم العامة

انتهت تمارين الأسبوع 5

جامعة
المنارة
HANARA UNIVERSITY