Practical Lecture : Type conversion

Quick Recap

Let’s take a quick recap of previous lecture –

A)

B)

C)

D)

E)

Today’s

Today we are going to cover -

- Basic concept of type conversion
- Type conversion- implicit and explicit
- Difference between implicit and explicit conversion
- Basic type to class type
- Class type to basic type
- One class to another class type

Let’s Get Started-

Type conversion

- Type conversion occur when there is a need to convert one data type to another.
- Mainly two types
    - Implicit – Also called type conversion
    - Explicit – Also called Type casting
- The basic difference between type conversion and type casting, i.e. type conversion is made “automatically” by compiler whereas, type casting is to be “explicitly done” by the programmer.

Type conversion in C

- Implicit conversion:
  - Implicit is automatic. Done by compiler without any extra trigger by the user
  - Implicit conversion done when more than one datatype present in the expression
  - All datatypes are upgraded to the datatype of largest variable.
  - It is possible for implicit conversions to lose information, signs can be lost (When signed is implicitly converted to unsigned), and overflow can occur (when long is implicitly converted to float).

Upgradation in implicit conversion example 1

Rule:

Bool-> char-> short int-> int-> unsigned int -> long-> unsigned-> long long -> float -> double -> long double

```
#include <iostream>
using namespace std;
int main() {
   int a = 10;
   char b = 'a';
   a = b + a;
   float c = a + 1.1;
   cout << "a : " << a << "\nb : " << b << "\nc : " << c;
}
```

Output:

```
a : 107
b : a
c : 108.1
```

Upgradation in implicit conversion example 1

In implicit conversion example given here, a is int type. When variable b is added to variable a (a=b+a), the ASCII value of variable b which is 97 is considered and added to 10 giving 107 in variable ‘a’. Here char is automatically upgraded to consider its int value (datatype of largest variable) .

Similarly for float, int is upgraded to float. So instead of considering value of a as 107, it considers as 107.0, added to 1.1 and then stored the result in C which is float.

Implicit conversion example 2

```
#include <iostream>
using namespace std;
int main() {
    int val1 = 11000;
    int val2 = 35600;
    long sum;
    sum = val1 + val2;
    cout << "val1 : " << val1 << "\nval2 : " << val2 << "\nsum : " << sum;
}
```

This program will not give an error but simply print the result.

Implicit conversion example 3

```
#include <iostream>
using namespace std;
int main()
{
   int x;
   for(x=97; x<=122; x++)
   {
        printf("%c", x); /*Implicit casting from int to char %c*/
   }
}
```

Explicit Type conversion

Also known as type casting and it is user defined

The user can typecast the result to make it of a particular data type.

This is done by using (type) operator

Syntax:

(type) expression

Before conversion, a runtime check is performed by compiler to see if destination can hold the source value.

Explicit Type conversion example 1

```
#include <iostream>
using namespace std;
int main()
{
    int a=5,b;
   float c=1.8;
   b=(int)(a+c);
  cout<<b<<endl;
   return 0;
}
```

Ouput : 6

Explicit Type conversion example 2

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

int main()
{
    double x = 1.2;
    // Explicit conversion from double to int
    int sum = (int)x + 1;
     cout << "Sum = " << sum;
     return 0;
}
```

Output: Sum=2

Difference between implicit conversion and Type casting

BASIS FOR COMPARISON	TYPE CASTING	TYPE CONVERSION
Meaning	One data type is assigned to another by the user, using a cast operator then it is called "Type Casting".	Conversion of one data type to another automatically by the compiler is called "Type Conversion".
Applied	Type casting can also be applied to two 'incompatible' data types.	Type conversion can only be implemented when two data types are 'compatible'.
Operator	For casting a data type to another, a casting operator '()' is required.	No operator required.
Size of Data Types	Destination type can be smaller than source type.	Here the destination type must be larger than source type.
Implemented	It is done during program designing.	It is done explicitly while compiling.
Conversion type	Narrowing conversion.	Widening conversion.

Example	int a;
	byte b;
	b= (byte) a;
		int a=3;
		float b;
		b=a; // value in b=3.000.

MCQ question

Which of the following is true?
The basic difference between type casting and type conversion is that

1. Type casting is done by the programmer.
2. Type conversion is done by the compiler while compiling.
3. Type conversion is done by the programmer.
4. Type casting is done by the compiler while compiling.

Options:

A. Only 1
B. Only 2
C. 1 & 2
D. 3 & 4

MCQ question

Which of the following is true?
The basic difference between type casting and type conversion is that

1. Type casting is done by the programmer.
2. Type conversion is done by the compiler while compiling.
3. Type conversion is done by the programmer.
4. Type casting is done by the compiler while compiling.

Options:

A. Only 1
B. Only 2
C. 1 & 2
D. 3 & 4

Answer: C

MCQ question

Which of the following is true?

1. Type casting can be applied to the datatypes, which may not be compatible with each other.
2. Type conversion can be applied to the datatypes which are compatible with each other.

Options:

A. Only 1
B. Only 2
C. 1 & 2
D. None of the above

MCQ question

Which of the following is true?

1. Type casting can be applied to the datatypes, which may not be compatible with each other.
2. Type conversion can be applied to the datatypes which are compatible with each other.

Options:

A. Only 1
B. Only 2
C. 1 & 2
D. None of the above

Answer: C

MCQ question

Which of the following is true?

1. In type casting, the destination type can be larger or smaller than the source type.
2. The destination type must be smaller than the source type in type conversion.

Options:

A. Only 1
B. Only 2
C. 1 & 2
D. None of the above

MCQ question

Which of the following is true?

1. In type casting, the destination type can be larger or smaller than the source type.
2. The destination type must be smaller than the source type in type conversion.

Options:

A. Only 1
B. Only 2
C. 1 & 2
D. None of the above

Answer: A

MCQ question

Type casting is called narrowing conversion while type conversion is called widening conversion.

Options:

A. True
B. False

MCQ question

Type casting is called narrowing conversion while type conversion is called widening conversion.

Options:

A. True
B. False

Answer: True

Difference between implicit conversion and Type casting

The basic difference is that type casting is done by the programmer. On the other hand, the type conversion is done by the compiler while compiling.

Type casting can be applied to the datatypes, which may not be compatible with each other. Conversely, type conversion can only be applied to the datatypes which are compatible with each other.

The conversion of one type to another in type casting requires the casting operator “( )” while the conversion of one data type to another in type conversion does not require any operator.

While converting one data type to another in type casting, the destination type can be larger or smaller than the source type. As against, the destination type must be larger than the source type in type conversion.

Difference between implicit conversion and Type casting

The conversion of one type to another type is done while coding in type casting. In contrast, in type conversion, the conversion of one type to another is done explicitly during compilation.

Type casting is called narrowing conversion because here the destination type can be smaller than source type. Unlike, type conversion is called widening conversion because here, the destination type must be larger than the source type.

Type conversion from user defined type to primary data type

We discussed type conversion when different types of constants and variables are used in expression

This automated type promotion will work well if both data types are of primary data type or both are of same user-defined data type.

But it will create problem when one data type is user-defined data type (like class) and another is primary data type.

For that we have to use some special function for type conversion as in such cases automatic type conversion can not be performed by the language itself.

Type conversion from user defined type to primary data type

There are three types of type conversion are possible:

1. Conversion from basic type to the class type.
2. Conversion from class type to basic type.
3. Conversion from one class to another class type.

Conversion from Basic type to the Class type

In this type of conversion the source type is basic type and the destination type is class type.

Means basic data type is converted into the class type.

For example we have class employee and one object of employee ‘emp’ and suppose we want to assign the employee code of employee ‘emp’ by any integer variable say ‘Ecode’ then the statement below is the example of the conversion from basic to class type.

```
emp = Ecode ;
```

Here the assignment will be done by converting “Ecode” which is of basic or primary data type into the class type.

Conversion from Basic type to the Class type

The conversion from basic type to the class type can be performed by two ways:

Using constructor

Using Operator Overloading

Conversion from Basic to the Class type using Constructor:

- We can use constructor to perform type conversion during the object creation.

- Consider the following example with class ‘Time’ in which we want to assign total time in minutes by integer variable ‘duration’.

- To achieve that we have implemented one constructor function which accepts one argument of type integer

- See the example on next slide

Practice Questions In Class

Write a program to convert basic type (duration in minutes) to class type (duration in hours and minutes) using constructor

```
#include <iostream>
using namespace std;
class Time {
	int hrs,min;
	public:
			Time(int);
		void display();
};
Time :: Time(int t) {
	cout<<"Basic Type to ==> Class Type Conversion..."<<endl;
	hrs=t/60;
	min=t%60;
}
```

Practice Questions In Class

```
void Time::display()
{
	cout<<hrs<< ": Hours(s)" <<endl;
	cout<<min<< " Minutes" <<endl;
}
int main()
{
	int duration;
	cout<<"Enter time duration in minutes : ";
	cin>>duration;
	Time t1=duration;
	t1.display();
	return 0;
}
```

Practice Questions In Class

Here, we have created an object “t1” of class “Time” and during the creation we have assigned integer variable “duration”.

It will pass time duration to the constructor function and assign to the “hrs” and “min” members of the class “Time”.

We have to note that during type conversion using the constructor we can pass only one argument and we can do type conversion at the type of initialization only.

Conversion from Basic to the Class type using operator overloading:

- We can also achieve type conversion by operator overloading.
- We can overload assignment operator for this purpose.
- Above example of Time class can be rewritten for type conversion using operator overloading concept to overload the assignment operator (=)
- By using overloaded assignment operator we can perform the type conversion at any place in program.
- See example on next slide

Practice Questions In Class

Write a program to convert basic type (duration in minutes) to class type (duration in hours and minutes) using operator overloading.

```
class Time {
	int hrs,min;
	public:
		void display();
		void operator=(int); // overloading function
};
void Time::display(){
	cout<<hrs<< ": Hour(s) "<<endl <<min<<": Minutes"<<endl ;
}
void Time::operator=(int t){
	cout<<"Basic Type to ==> Class Type Conversion..."<<endl;
	hrs=t/60;
	min=t%60;
}
```

Practice Questions In Class

```
int main()
{
      Time t1;
      int duration;
      cout<<"Enter time duration in minutes";
      cin>>duration;
      cout<<"object t1 overloaded assignment..."<<endl;
      t1=duration;
      t1.display();
      cout<<"object t1 assignment operator 2nd method..."<<endl;
      t1.operator=(duration);
      t1.display();
      return 0;
}
```

Assignment

Write a program to create a class employee and one object of employee ‘emp’ . Get an employee code from user in variable Ecode. Assign the employee code of employee ‘emp’ by any integer variable say ‘Ecode’ so as to do conversion from basic to class type as follows.

```
emp = Ecode ;
```

Implement the above program using both the methods:

1. using constructor
2. using operator overloading

MCQ question

Which of the following options are true?
The conversion from basic type to the class type can be performed by two ways:

1. Using constructor
2. Using Operator Overloading

Options:

A. 1 only
B. 2 only
C. None of the above
D. 1 & 2 both

MCQ question

Which of the following options are true?
The conversion from basic type to the class type can be performed by two ways:

1. Using constructor
2. Using Operator Overloading

Options:

A. 1 only
B. 2 only
C. None of the above
D. 1 & 2 both

Answer: D

MCQ question

Which of the following is true?

1. During type conversion using the constructor we can pass only one argument .
2. we can do type conversion at the type of initialization only.

Options:

A. 1&2

B. None of the both

C. 1 only

D. 2 only

MCQ question

Which of the following is true?

1. During type conversion using the constructor we can pass only one argument .
2. we can do type conversion at the type of initialization only.

Options:

A. 1&2
B. None of the both
C. 1 only
D. 2 only

Answer: option A

Type conversion from class type to basic type

In this type of conversion the source type is class type and the destination type is basic type.

Means class data type is converted into the basic type.

For example we have class Time and one object of Time class ‘t’ and suppose we want to assign the total time of object ‘t’ to any integer variable say ‘duration’ then the statement below is the example of the conversion from class to basic type.

duration= t ; // where, t is object and duration is of basic data type

Here the assignment will be done by converting “t” object which is of class type into the basic or primary data type.

Type conversion from class type to basic type

It requires special casting operator function for class type to basic type conversion.

This is known as the conversion function.

The syntax for the conversion function is as under:

```
operator typename( )
{
          ....
      ....
}
```

Practice Questions In Class

Write a program in C++ to assign time in hours and minutes in the form of total time in minutes into one integer variable “duration” using both the methods.

1. Using constructor

2. Using operator overloading

Practice Questions In Class

```
#include <iostream>
using namespace std;
class Time
{
        int hrs,min;
        public:
                Time(int ,int);    // constructor
                operator int();    // casting operator function
                ~Time()            // destructor
                {
                        cout<<"Destructor called..."<<endl;
                }
};
```

Practice Questions In Class

```
Time::Time(int a,int b)
{
	cout<<"Constructor called with two parameters..."<<endl;
	hrs=a;
	min=b;
}

Time :: operator int()
{
	cout<<"Class Type to Basic Type Conversion..."<<endl;
	return(hrs*60+min);
}
```

Practice Questions In Class

```
int main(){
	int h,m,duration;
	cout<<"Enter Hours ";
	cin>>h;
	cout<<"Enter Minutes ";
	cin>>m;
	Time t(h,m);	// construct object
	duration = t;	// casting conversion OR duration = (int)t
	cout<<"Total Minutes are "<<duration <<endl;
	cout<<"2nd method operator overloading "<<endl;
	duration = t.operator int();
	cout<<"Total Minutes are "<<duration <<endl;
	return 0;
}
```

Practice Questions In Class

Output:

```
Enter Hours 2

Enter Minutes 45

Constructor called with two parameters...

Class Type to Basic Type Conversion...

Total Minutes are 165

2nd method operator overloading

Class Type to Basic Type Conversion...

Total Minutes are 165
```

Type conversion from class type to basic type

Notice the statement in above program where conversion took place.

```
      duration = t;
```

We can also specify the casting type and write the same statement by the following way to achieve the same result.

```
    duration = (int) t;          // Casting
```

The conversion function should satisfy the following condition:

- **It must be a class member.**
- **It must not specify the return value even though it returns the value.**
- **It must not have any argument.**

Assignment

Write a program to create a class employee and one object of employee 'emp' . Assign some salary to employee's salary in emp object. Get a float value in a variable called payment. Assign this employee salary to variable payment so as to do conversion from class type to basic type as follows.

```
payment= emp ;
or
payment= (float) emp;
```

Implement the above program using both the methods:

1. using constructor
2. using operator overloading

MCQ question

Special casting operator function for class type to basic type conversion is known as the __________ function.

MCQ question

Special casting operator function for class type to basic type conversion is known as the conversion function.

MCQ question

Which of the following is false:
The conversion function should satisfy the following condition:
1. It must be a class member.
2. It must specify the return value even though it returns the value.
3. It must have any argument.

Options:
A. 1 & 3
B. 1 & 2
C. 2 & 3
D. 1,2 & 3

MCQ question

Which of the following is false:
The conversion function should satisfy the following condition:

1. It must be a class member.
2. It must specify the return value even though it returns the value.
3. It must have any argument.

Options:

A. 1 & 3
B. 1 & 2
C. 2 & 3
D. 1,2 & 3

Options: option C

Conversion from one class type to another class type

In this type of conversion both the type that is source type and the destination type are of class type.

Means the source type is of class type and the destination type is also of the class type.

In other words, one class data type is converted into the another class type.

Conversion from one class to another class can be performed either by using the constructor or type conversion function.

Conversion from one class type to another class type

For example we have two classes one for “computer” and another for “mobile”.

Suppose if we wish to assign “price” of computer to mobile then it can be achieved by the statement below which is the example of the conversion from one class to another class type.

```
    mob = comp ;
```

// where mob and comp are the objects of mobile and computer classes respectively.

Here the assignment will be done by converting “comp” object which is of class type into the “mob” which is another class data type.

Practice Questions In Class

Write a C++ program that convert class Time to another class Minute demonstrating conversion from one class to another class type. Overload = operator for conversion purpose.

Hint: Declare two classes *“Time”* and *“Minute”* respectively. Create objects of the same. Assign one object to another.

Practice Questions In Class

```
#include <iostream>
using namespace std;
class Time
{
	int hrs,min;
	public:
	Time(int h,int m)
	{
		hrs=h;
		min=m;
	}
	Time()
	{
		cout<<"\n Time's Object Created";
	}
```

Practice Questions In Class

```
int getMinutes()
{
    int tot_min = ( hrs * 60 ) + min ;
        return tot_min;
}
void display()
{
    cout<<"Hours: "<<hrs<<endl ;
        cout<<" Minutes : "<<min <<endl ;
}
};
```

Practice Questions In Class

```
class Minute
{
      int min;
      public:
      Minute()
 {
           min = 0;
 }
void operator=(Time T)
{
      min=T.getMinutes();
}
void display(){
      cout<<"\n Total Minutes : " <<min<<endl;
}
};
```

Practice Questions In Class

Output:

```
Hours: 2
Minutes : 30

Total Minutes : 0

Hours: 2
  Minutes : 30
    Total Minutes : 150
```

Assignment

Define two classes one for “*computer*” and another for “*mobile*”*. Let us have attributes like model, price etc. A*ssign *“price”* of *computer* to *mobile* *using* the statement below which is the example of the conversion from one class to another class type.
mob = comp ; // where mob and comp are the objects of mobile and computer classes respectively. Here the assignment will be done by converting *“comp”* object which is of class type into the *“mob”* which is another class data type.

Implement the above code by overloading = operator.

MCQ question

State true or false.
Conversion from class type to class type can be done only using operator overloading .

Options:
True
False

MCQ question

State true or false.
Conversion from class type to class type can be done only using operator overloading .

Options:
True
False

Answer: False

Assignment : Implement the program of time and minutes to convert one class into another using constructor.

MCQ question

Choose the correct option:

Conversion from one class to another class can be performed by

1. using the constructor .
2. using type conversion function.
3. using operator overloading .
4. Using ‘=‘ operator which is a Conversion function

Options:

A. 1,2
B. 1,2,3
C. 1,2,3,4
D. 2,3

MCQ question

Choose the correct option:

Conversion from one class to another class can be performed by

1. using the constructor .
2. using type conversion function.
3. using operator overloading .
4. Using ‘=‘ operator which is a Conversion function

Options:

A. 1,2
B. 1,2,3
C. 1,2,3,4
D. 2,3

Answer: Option C

Any Questions ??

Thank You!

See you guys in next class.