

RTTI

Contents

- What is RTTI
- typeid operator
- Need of dynamic_cast.
- reinterpret_cast

RTTI(Run Time Type Identification)

- Type checking is required at two points of time in a program.
 1. Compile time(known as compile time checking)
 2. Run time(known as dynamic type checking)
- RTTI enables identifying the type of an object during execution of program.
- There are two types of RTTI operators.
 1. typeid operator
 2. dynamic_cast operator.
- RTTI operators are runtime events for polymorphic classes and compile time events for all other types.

typeid operator

- Allows the type of an object to be determined at run time.
- Typeid returns the type of user defined and built in data types.
- Need to include <typeinfo> header file.
- Two functions are used
 1. name()
 2. raw_name()

typeid example

```
int num1;
```

```
float num2;
```

```
cDate d1;
```

```
cout<<"Type of num1 is= "<<typeid(num1).name(); //int
```

```
cout<<"Type of num1 is= "<<typeid(num2).name(); //float
```

```
cout<<"Type of num1 is= "<<typeid(d1).name(); //class cDate
```

Need of dynamic_cast

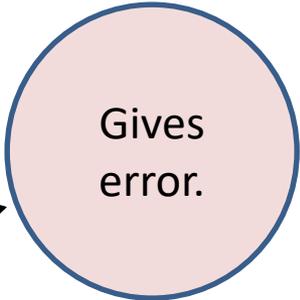
```
class BaseClass
{
    public:
        virtual void show()
        {
            cout<<"Base class";
        }
};

class DerivedClass : public BaseClass
{
    public:
        void show()
        {
            cout<<"Derived class";
        }
        void display()
        {
            cout<<"Specific fun";
        }
};
```

```
int main( )
{
    BaseClass* ptr;
    DerivedClass d1;

    ptr=&d1;

    ptr -> display( );
}
```



Gives error.

dynamic_cast operator

- Performs safe down casting.
- Works with pointers only and not with objects.

```
int main()
```

```
{
```

```
    BaseClass* ptr;
```

```
    DerivedClass d1;
```

```
    ptr = &d1;
```

```
    ptr->show();
```



Virtual
function

```
    DerivedClass *p = dynamic_cast<DerivedClass*>(ptr);
```

```
    p->display();
```



Non
virtual
function

```
}
```

reinterpret_cast

- Performs low level interpretation of bit pattern . It allows to edit individual bytes in memory by using bitwise/bitshift operators.

For example:- Writing linkers or loaders, it may be necessary to edit instructions.

- Used to convert any data type to any other data type.
- For e.g.

```
cComplex pcom;
```

```
char *pc = reinterpret_cast <char*> (pcom);
```

Lab assignments

- Modify the cEmployee hierarchy to add a function bonus() in the cManager class. Bonus is given only to manager and other employees get their normal salary.
- Use typeid operator to find the type of objects at run time.