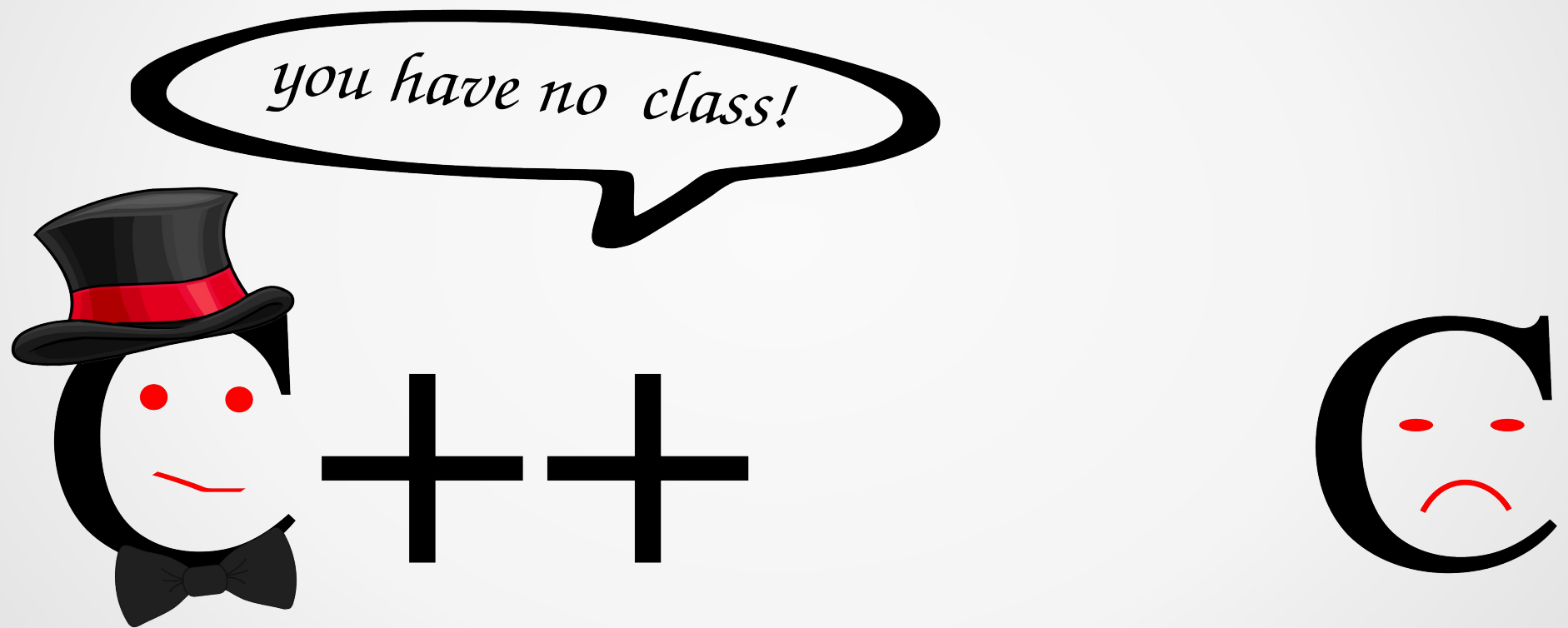


C++ Classes (Part I)



Krishna Kumar

Class Basics

- A class is a user-defined type.
- A class consists of a set of members. The most common kinds of members are data members and member functions.
- Member functions can define the meaning of initialization (creation), copy, move, and cleanup (destruction).
- Members are accessed using . (dot) for objects and → (arrow) for pointers.
- Operators, such as +, !, and [], can be defined for a class.
- A class is a namespace containing its members.
- The public members provide the class's interface and the private members provide implementation details

Class basics

```
class X {  
    private:                                // the representation (implementation) is private  
        int m;  
    public:                                // the user interface is public  
        X(int i =0) :m{i} { }             // a constructor (initialize the data member m)  
  
        int mf(int i)                      // a member function  
        {  
            int old = m;  
            m = i;                          // set a new value  
            return old;                     // return the old value  
        }  
};  
  
X var {7}; // a variable of type X, initialized to 7  
  
int user(X var, X* ptr)  
{  
    int x = var.mf(7);                      // access using . (dot)  
    int y = ptr->mf(9);                     // access using -> (arrow)  
    int z = var.m;                          // error: cannot access private member  
}
```

Constructor

```
class Date {  
    int d, m, y;  
public:  
    // ...  
  
    Date(int, int, int);           // day, month, year  
    Date(int, int);               // day, month, today's year  
    Date(int);                   // day, today's month and year  
    Date();                      // default Date: today  
    Date(const char*);           // date in string representation  
};  
  
Date today {4};                 // 4, today.m, today.y  
Date july4 {"July 4, 1983"};  
Date guy {5,11};               // 5, November, today.y  
Date now;                      // default initialized as today  
Date start {};
```

- declare a function with the explicit purpose of initializing objects.
- Such a function constructs values of a given type, it is called a constructor.
- A constructor is recognized by having the same name as the class itself.
- Use {} to represent intialisation over ().
- By guaranteeing proper initialization of objects, the constructors greatly simplify the implementation of member functions

References

-
- Exceptional C++ - Herb Sutter