OBJECT ORIENTED PROGRAMMING

Polymorphism

Zuhaib A. Shaikh,
Asst. Prof., CSE Deptt.,QUEST
Web: zuhaib-shaikh.neocities.org

Introduction

- Poly = multiple, morphism=shapes/definitions
- Allows to define single entity many times
- Entity types:
  - Functions
  - Operators
- Types
  - Overloading, also known as compile-time polymorphism
  - Overriding, also known as run-time polymorphism
- It provides
  - An ease to programmers to use the same name for multiple definitions
  - Facility of reusability at some extent specially in inheritance
  - An aid to create and use new data types

Overloading

- Type of polymorphism
- A single definition from many definitions is called on basis of arguments
- Definitions are differentiated w.r.t number, types and arrangement of arguments
- This type of polymorphism is known before compilation
- Function Overloading
  - A function with same name is declared and defined many times
  - Each declaration and its definition should differ in arguments
  - During function call, only corresponding definition w.r.t. argument(s) is called
  - Constructor Overloading
    - A constructor can also be overloaded by differencing in the arguments
    - Provides a facility to create objects of a class in different ways
    - Also allow to create temporary objects

Overloading

- Operator Overloading
  - The definition of an operator can be represented as function definition starting with keyword *"operator"*
  - Requires object-class structure of program
  - Types:
    - Unary operator overloading
    - Binary operator overloading
  - Unary operator
    - Those operators which operate single operand
    - e.g. increment or decrement operator
    - The unary operator can be used and declared as
      - Postfix: After the operand
      - Prefix: Before the operand
    - The prefix or postfix operator definition is differentiated by writing *"int"* in the argument of function or leaving those empty respectively

Overloading

- Operator Overloading
  - A temporary object can be created and returned with help of a constructor
  - Temporary object provide a facility to assign the attributes to another object as return value
  - If there isn't any return handler, then temporary object will be discarded
  - Binary operator
    - Those operators which operate two operands
    - e.g. arithmetic operators
    - Binary operator always has an argument of the same class and optional return value
    - The object on the left of the operator is calling object
    - The object on the right side of operator is considered as an argument

Overriding

- Can only be implemented with help of inheritance
- A method in parent class is defined in child class with same name and parameters
- Run-time polymorphism
- Helpful to override new method on existing one without replacing it
- Method belongs to appropriate class will be called only
- Entities to override:
  - Methods
  - Operators
- Overriding sometimes lead us to “Diamond Problem”

Object type conversion

- Objects can be converted from one type to another
- Existing casting methods can be overloaded
- Casting can be overloaded using operator keyword
- Syntax

```
class A
{
private:
        members:
 public:
        members:
};
```

```
class B
 {
private:
        members;
public:
        operator A() const
        {
                return A();
        }
};
```

- Objects of class B can be casted automatically for objects of class A.

Questions