

Basic Concept of Object Oriented and Procedure Oriented Programming

Mala Dutta

Assistant Professor

Department of computer Engineering

IET DAVV

Indore India

ABSTRACT

This paper discusses about basic concept of Procedure Oriented Programming and Object Oriented Programming. Procedure oriented programming means “set of procedure”, which is a “set of functions”. Functions are called repeatedly in a program to execute tasks performed by them. The core of the pure object-oriented programming is to create an object, in code, that has certain properties and methods. While designing modules, we try to see whole world in the form of objects.

Keywords:- Field modifiers, Access control modifiers, Class members, final modifier, Method invocation, Method declaration, arguments.

I. INTRODUCTION

A **class** is a blueprint of an object. You can think of a class as a concept, and the object is the embodiment of that concept. You need to have a class before you can create an object. So, let's say you want to use a person in your program. You want to be able to describe the person and have the person do something. A class called 'person' would provide a blueprint for what a person looks like and what a person can do. To actually use a person in your program you need to create an object. You use the person class to create an object of the type 'person.' Now you can describe this person and have it do something.

A **function** is a combination of instructions that are combined to achieve some result. A function typically requires some input (called arguments) and returns some results. For example, consider the example of driving a car. To determine the mileage, you need to perform a calculation using the distance driven and the amount of fuel used. You could write a function to do this calculation. The arguments going into the function would be distance and fuel consumption, and the result would be mileage. Anytime you want to determine the mileage, you simply call the function to perform the calculation.

How does this differ from a method? A function is independent and not associated with a class. You can use this function

anywhere in your code, and you don't need to have an object to use it.

C++ is a hybrid computer programming language, which includes the characteristics of procedure-based and object-oriented. Most studies in the comparison of procedure-based programming and object-oriented languages are focused on the algorithm or scientific applications. In this paper we strive to compare the characteristics of the object-oriented and procedure-based language with implications for business applications. Data anchor and structures applications in the business information system design are demonstrated and also are used to compare the performance between procedure-based and object-oriented languages. In this experiment we will test only basic operations and not the performance of complex systems such as the speed of reading and modifying data. The outcome of this study will serve as a future technical analysis reference for software business application in designing information systems.

II. PROCEDURE ORIENTED PROGRAMMING

Procedure oriented programming means “set of procedure” , which is a “set of functions”. functions are called repeatedly in a program to execute tasks performed by them ,for example : A program may involve collecting data from user (reading), performing some kind of calculations on the data, and

finally displaying the result to the user. All three tasks can be written in a program with the help of three functions which perform three different tasks.

Object-Oriented Programming: The core of the pure object-oriented programming is to create an object, in code, that has certain properties and methods. While designing modules, we try to see the whole world in the form of objects.

For example, a car is an object which has certain properties such as color, number of doors, engine size, transmission type, number of doors, etc. It also has certain methods such as accelerate, brake, and so on.

Procedural vs. Object-Oriented Programming:

The unit in procedural programming is *function*, and the unit in object-oriented programming is *class*.

Procedural programming concentrates on creating functions, while object-oriented programming starts from isolating the classes, and then look for the methods inside them.

Procedural programming separates the data of the program from the operations that manipulate the data, while object-oriented programming focuses on both of them.

In procedure-oriented programming, emphasis is on procedure abstraction and follows a top-down approach, while in object-oriented programming, emphasis is on data abstraction and follows a bottom-up approach.

Procedure-oriented is suitable for small programs, while object-oriented is suitable for large programs.

Procedure-oriented programs are divided into small parts called functions, while in object-oriented programs are divided into parts called objects.

POP does not have any proper way for hiding data so it is less secure. OOP provides data hiding so it provides more security. Examples of POP are: C, VB, FORTRAN, and Pascal. Examples of OOP are: C++, JAVA.

Concept of Class and Object:

Class refers to a blueprint. It defines the variables and methods the objects support.

“Object” is an instance of a class. Each object has a class which defines its data and behavior.

Class Members: A class can have three kinds of members:

fields: data variables which determine the status of the class or an object

methods: executable code of the class built from statements. It allows us to manipulate/change the status of an object or access the value of the data member.

nested classes and nested interfaces.

Sample class

```
class Pencil {
    public String color = "red";
    public int length;
    public float diameter;
    public static long nextID = 0;
    public void setColor (String newColor) {
        color = newColor;
    }
}
```

Fields Declaration

a type name followed by the field name, and optionally an initialization clause

Primitive data type vs. Object reference

boolean, char, byte, short, int, long, float, double

field declarations can be preceded by different modifiers

access control modifiers: static, final.

More about field modifiers and Access control modifiers:

Private: private members are accessible only in the class itself.

Package: package members are accessible in classes in the same package and the class itself.

Protected: protected members are accessible in classes in the same package, in subclasses of the class, and in the class itself.

Public: public members are accessible anywhere the class is accessible.

More about field modifiers static:

Only one copy of the static field exists, shared by all objects of this class. It can be accessed directly in the class itself. Access from outside the class must be preceded by the class name.

or via an object belonging to the class from outside the class, non-static fields must be accessed through an object reference.

```
public class CreatePencil {
    public static void main (String
    args[]){
        Pencil p1 = new Pencil();
        Pencil.nextID++;

        System.out.println(p1.nextID);
        //Result?
        Pencil p2 = new Pencil();
        Pencil.nextID++;

        System.out.println(p2.nextID);
        //Result?

        System.out.println(p1.nextID);
        //Result?
    }
}
```

Note: this code is only for the purpose of showing the usage of static fields. It has POOR design!

final modifier:

In finalmodifier once initialized, the value cannot be changedoften be used to define named constantsstatic final fields must be initialized when the class is initializednon-static final fields must be initialized when an object of the class is constructed.

Field initialization:

Fields Initializationnot necessary to be constants, as long as with the right type.If no initialization, then a default initial value is assigned depending on its type,

Type	Initial Value
boolean	false
char	'\u0000'
byte, short, int, long	0
float	+0.0f
double	+0.0
object reference	null

Methods – Declaration

Method declaration: two parts

1. Method header:consists of modifiers (optional), return type, method name, parameter list and a throws clause (optional)

Types of modifiers

access control modifiers

abstract

the method body is empty. E.g.abstract void sampleMethod();

static

represent the whole class, no a specific objectcan only access static fields and other static methods of the same class

final

Cannot be overridden in subclasses

2. Method body

Methods – Invocation

Method invocations invoked as operations on objects/classes using the dot (.)

Operator.

Reference.method(arguments)

static method:

Outside of the class: “reference” can either be the class name or an object reference belonging to the class

Inside the class: “reference” can be omitted non-static method:

“reference” must be an object reference.

Method – Overloading

A class can have more than one method with the same name as long as they have different parameter list.

```
public class Pencil {
    ...
    public void setPrice(float
    newPrice) {
        price = newPrice;
    }
    public void setPrice(Pencil p)
    {
        price = p.getPrice();
    }
}
```

```
}
}
```

How does the compiler know which method you're invoking? — compares the number and type of the parameters and uses the matched one

Methods – Parameter Values

Parameters are always passed by value.

```
public void method1 (int a) {
    a = 6;
}

public void method2 ( ) {
    int b = 3;
    method1(b);    // now b = ?
                  // b = 3
}
```

When the parameter is an object reference, it is the object reference, not the object itself, getting passed.

The Main Method – Concept

Main method

The system locates and runs the main method for a class when you run a program other methods get execution when called by the main method explicitly or implicitly must be public, static and void.

The Main Method - Getting Input from the Command Line

When running a program through the java command, you can provide a list of strings as the real arguments for the main method. In the main method, you can use args[index] to fetch the corresponding argument

```
class Greetings {
    public static void main (String args[]){
        String name1 = args[0];
        String name2 = args[1];

        System.out.println("Hello " + name1
            + "&" + name2);
    }
}
```

Modifiers of the classes

A class can also has modifiers

Public: publicly accessible. Without this modifier, a class is only accessible within its own package.

Abstract: no objects of abstract classes can be created all of its abstract methods must be implemented by its subclass; otherwise that subclass must be declared abstract also.

Final: cannot be sub-classed. We cannot extend the class if final modifier is used in class.

III. CONCLUSION

The purpose of this paper is to find approach and use of procedure and object oriented programming. Reusability is one of the quality of the object oriented programming. In procedure oriented programming functions are called repeatedly in a program to execute tasks performed by them. The core of the pure object-oriented programming is to create an object, in code, that has certain properties and methods.

REFERENCES

- [1] https://www.google.co.in/?gfe_rd=cr&ei=us1CV8mwFejI8AeXhrjQCg&gws_rd=ssl#q=ppt+on+object+oriented+programming+concepts
- [2] <http://www.slideshare.net/HarisBinZahid/procedural-vs-object-oriented-programming>