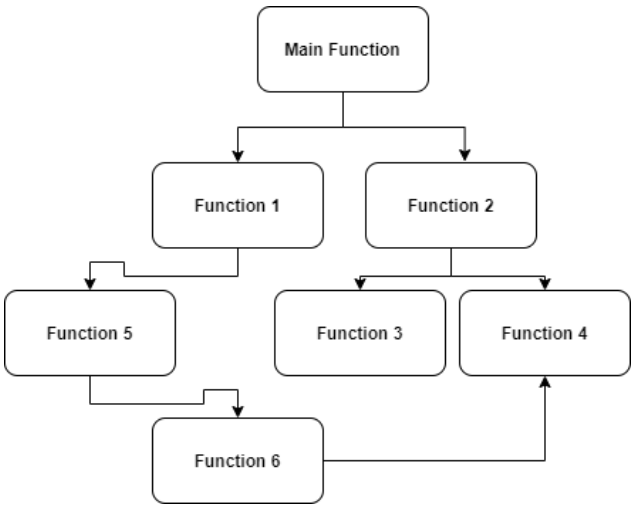
OOPS Concepts

Table Of Contents

A Complex Problem Dividing



Breaking a complex program into simple one

- OOP is build upon the reference of structured programming and data abstraction

▼ **Simple Example**

```
package Object_Oriented_Concepts._1_ClassAndObject;

import org.omg.CORBA.DynAnyPackage.Invalid;

import java.util.Scanner;

//class to print a to z characters
class printAZ
{
    char c;
    public void operation()
    {
        for(c= 'a';c<='z';c++)
        {
            System.out.print(c+" ");
        }
    }
}

// class to check the string is palindrome or not
class palindrome
{
    String data;
```

```
    public void palindromeOperation()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String data = scanner.next();
        String revdata = "";
        for(int i = data.length()-1;i>=0;i--)
        {
            revdata = revdata+data.charAt(i);
        }

        if(data.equals(revdata))
        {
            System.out.println("String "+data+" is palindrome");
        }
        else
        {
            System.out.println("String "+data+" is not palindrome");
        }

    }
}

// class to print the reverse of string
class reverseString
{
    String normalStr;
    public void reverseOperation()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String normalStr = scanner.next();

        String reverseStr = "";
        for(int i = normalStr.length()-1;i>=0;i--)
        {
            reverseStr = reverseStr+normalStr.charAt(i);
        }
        System.out.println("The normal string is: "+normalStr+" and the reverse string is: "+reverseStr);
    }
}

// class to print the multiplication table
class multiply
{
    int num;

    public void multiplyOperation()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number: ");
        num = scanner.nextInt();

        for(int i = num; i<=10;i++)
        {
            System.out.println(num+"*"+i+"="+num*i);
        }
    }
}


public class _1_simpleProgram_Example
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter which logic to perform " +
                "\n1: printing A - Z" +
                "\n2: checking palindrome" +
                "\n3: reversing the string" +
                "\n4: multiplication table" +
```

```
                "\nPlease select your choice: ");
        int choice = scanner.nextInt();
        switch(choice)
        {
            case 1:
                printAZ c = new printAZ();
                c.operation();
                break;
            case 2:
                palindrome pc = new palindrome();
                pc.palindromeOperation();
                break;
            case 3:
                reverseString rs = new reverseString();
                rs.reverseOperation();
                break;
            case 4:
                multiply mp = new multiply();
                mp.multiplyOperation();
                break;
            default:
                System.out.println("Invalid choice, Please choose from 1 to 4.....!");
        }
    }
}
//output
Enter which logic to perform
1: printing A - Z
2: checking palindrome
3: reversing the string
4: multiplication table
Please select your choice: 1
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

Important Terms

- Important terms in OOPS concept

1. **Class :** It is the collection of **Data Members & Member Functions and** class is a blueprint/template which is used to create a **object**

    - Syntax Of Class

```
SYNTAX OF THE CLASS:
--------------------
  access modifier Class Classname
      {
        variable
        methods
        blocks
        constructors
        interface
        nested class
        abstraction
      }

  SYNTAX OF A METHOD:
  -------------------
  access modifier returnType methodName (parameter)
    {
    ---------------
    ---------------   --> body
    ---------------
    }
```

- Simple Structure To Show What Is Class

```
Class Lion
{
  String name;
  String breed;   // has part of the object
  String gender;


  eat()
  {
      ------
      ------
  }

  sleep()
  {
      -------
      -------
  }

  run()
  {
      -------
      -------
  }

}
```

2. Object : it is the **instance of class**

   a. Creating Of Object Ways - In java we have 5 ways to create an object for a class, in most of the time we use **new** **Keyword to create an object like this** >>> "new class_name()"

   ```
   1. using 'new' keywords
   2. using newInstance()
   3. using clone()
   4. using FactoryMethod
   5. using deserialization
   ```

   b. Creating an object using "new"

   In order to create an object we need to follow 3 steps

   ```
   CREATION OF OBJECT USING 'new' KEYWORD:
   ---------------------------------------

   1. Declaration: declare a variable to a type(class) eg: Lion l1
   2. Instantiation: allocate the memory (new) memory is pointed by reference variable  eg: new Lion
   ();
   3. Initilization : variables and methods will be loaded to the memory.
   ```

   after you creating an object it look like this

   ```
   //syntax
   //class_name variable = new class_name();
   Lion l1 = new Lion();
   ```

   c. Simple Example Of Creating class and object for lion

```
package _6OOPSConcepts;

public class _1oopsEx1
{
    public static void main(String[] args)
    {
        //this is main class

        //creating an object to a class lion

        lion l1 = new lion(); //SYNTAX:  class_name variable = new class_name();

        //calling member functions of the lion class
        l1.weight();
        l1.height();
        l1.color();

    }
}
class lion
{

    //these are data members
    String weight_lion = "120kg";
    String height_lion = "200cm";
    String color_lion = "Orange-Brown";

    //these are member functions
    void weight()
    {
        System.out.println("The lion's weight is around: "+weight_lion);
    }
    void height()
    {
        System.out.println("The lion's height is around: "+height_lion);
    }
    void color()
    {
        System.out.println("The lion's color is: "+color_lion);
    }
}
//output
The lion's weight is around: 120kg
The lion's height is around: 200cm
The lion's color is: Orange-Brown

Process finished with exit code 0
```

NOTE:

- Always program execution starts from the class where main method is present.
- To access the variables and methods present in different class we will make use of reference variable.

d. Object value can be initialized using 3 ways

```
OBJECT VALUES CAN BE INITILIZED USING 3 WAYS:
---------------------------------------------
1. using reference varibale
2. using methods
3. using the constructor
```

1. Initializing object value using reference variable

```
package _6OOPSConcepts;

//Initializing object value using reference variable
public class _2oopsEx2
{
    public static void main(String[] args)
    {
        //Object creation
        car c1 = new car();

        // values for objects is given using ref variables
        c1.carBrand = "TATA";
        c1.carName = "RANGE ROVER";
        c1.carColor = "Red";

        // methods are called with the help of ref variable.
        c1.carselected();
        c1.carInteriorFeature();
        c1.safetyFeatures();
        c1.carDimensions();

    }
}

class car
{
    String carBrand;
    String carName;
    String carColor;

    void carselected()
    {
        System.out.println("Selected Car");
        System.out.println("Car Name: "+carName);
        System.out.println("Car Brand: "+carBrand);
        System.out.println("Car Color: "+carColor);
        System.out.println();
    }
    void carInteriorFeature()
    {
        System.out.println("The car "+carName+" from "+carBrand+" has ton of interior features
like\n1.instrumental cluster\n2.digital connections(apple carplay, android auto\n3.bose sound
 system");
        System.out.println();
    }
    void safetyFeatures()
    {
        System.out.println("The car "+carName+" has\n1.4 Airbags\n2.EBS Breaking System\n3.Cru
ise control\n4. 4 sides disc break ");
        System.out.println();
    }
    void carDimensions()
    {
        System.out.println("The car "+carName+" has\n1. width = 4 meters\n2. height = 5.4inc
h");
        System.out.println();
    }

}
//output
Selected Car
Car Name: RANGE ROVER
Car Brand: TATA
Car Color: Red

The car RANGE ROVER from TATA has ton of interior features like
1.instrumental cluster
2.digital connections(apple carplay, android auto
3.bose sound system
```

```
The car RANGE ROVER has
1.4 Airbags
2.EBS Breaking System
3.Cruise control
4. 4 sides disc break

The car RANGE ROVER has
1. width = 4 meters
2. height = 5.4inch


Process finished with exit code 0
```