23

Assignment - 3

By.

K. Senthamizhselvan,

192321143.

DBMS-0985

Final

The final keyword is a non-access Modifier used for classes, attributes and Methods. Which Makes them non-changable.

example

```
class A {
    Final int A = 100;
          int B = 0;
Public void disp1()
  {
    B = B+1;
System.out.Println("b="+B);
Public Void disp(){
System.out.Println("a="+A);
}
Public static void Main (String[] args){
    A A1 = new A();
    A A2 = new A();
     A1.disp
     A2.disp
```

Out Put

```
b = 0
a = 100
b = 0
a = 100,
```

Try - Catch

The try statement allows you to define a block of code to be tested for errors while it is being executed.

example

```
Public class Main{
    Public static void Main(String[] args){
    try
    {int a = 1/100;
    System.out.println("rest of the code in try block");
    }
    Catch (ArrayIndexOutOfBoundsException e){System.out.println("Array Index out of bound exception");}
    }
```

output

rest of the code in try block.

Bound

The Index out of Bounds Exception is a common issue in Java that occurs when trying to access elements at invalid indices in array, list or other data structure.

example:

```
Public class Main
  {
  Public static void main (String [] args)
  {
try
{ int a[] = {1, 2, 3};
  system.out.println (a [i]);
catch (Array Index out of Bound Exception e);
{
System.out.println("Array Index out of Bound"
                    + e.getmessage);
}
}
```

Math

The Java Math class has many methods that allow you to perform mathematical task on numbers.

example

```
Public class Main{
    Public static void main(string[] args){
    int a = 30;
system.out.Println("sign value : " + Math.Sin(a
system.out.Println("cosin value : " + Math.cos(a)
```

Arraylist

Arraylist is a Java class implemented using the List Interface.

```
import java.util.ArrayList;
Public class Main{
  Public static void Main (string[] args){

ArrayList<string> cars = new ArrayList<string>();

    cars.add("Volvo");
    cars.add("Bow");
    cars.add("Ford");

  System.out.println(cars);
  }
}
```

Output

[Volvo, Bow, Ford, cars]

Type casting

```
Public class main
    Public static void main (String[] args)
    int a = 10;
    double b = a;
    float c = 100;
    float d = c;
     byte e = 11;
     long f = e
    char ch = "A"
        int g = ch;
    }
    System.out.println(b);
    system.out.println(d);
    system.out.println(e);
    system.out.println(g);
}
}
```

Output.

10.0
100.0
11
65

Break and Continue

```
class A {
  Public static void main (String [] args)
  {
    int i;
    for (i=0; i<10; i++) {
      if (i==5)
        break;
      System.out.println ("i:" +i);
    }
    System.out.println ("out of loop");
  }
}
```

Out Put;

i:0
i:1
i:2
i:3
i:4
out of Loop.

Generic class

A generic class is defined just like a Normal class. With the addition of the list of the type Parameters in angle brackets <...> after the class Name.

example

```
Class A<T>
{
Private T data;
 A(T data){
 this.data =data;
}
 Public T. getdata()
 {
  return this. data;
}
}
Class Main {
   Public static void Main (String[] args) {
   A <Integer> obj1 = new A<>(5);
System. out. Println ("Generic class returned:" + obj1. getdata());
A <String> obj2 = new A<>("Java Programming");
System. out. Println ("Generic class returned:" +obj2. getdata());
}
}
```

Array List

java Arraylist is a Part of java collection framework and it is a class of java.util.Package. it Provides us with dynamic array in java.

example.

```
Import java.util.ArrayList;
Public class Main{
    Public static void Main (String [] args) {
        obj.add ("One");
        obj.add ("Two");
        obj.add ("Three");
    System.out.Println ("Array List" + obj);
    }
}
```

Output

Array List [One, two, three]

Stack

Stack is a Linear data structure that follows the LIFO (last in first out) principle

```
import java.lang.*;
import java.util.stack;

class Main
{ public static void main (String[] args){
fruits.Push ("APPle");
fruits.Push ("Orange");
fruits.Push ("Mango");

System.out.println ("stack : " +fruits);
}
}
```

Output

APPle

Orange

Mango

Queue

Queue is a Linear data structure that follow FIFO.

example

```
Import java.util.LinkedList;
Import java.lang.*;
Import java.util.Queue;

class Main
{
Public static void Main (String[] args)
{
Queue <String> fruit = new LinkedList<>();
    fruits.add ('APPle');
    fruits.add ('Orange');
    fruits.add ('Mango');
  System.Out.Println ("Queue:" + fruit);
}
}
```

OutPut

Queue: [APPle, Orange, Mango]

Hash Map

Hash Map in java stores the data in (Key, Value) pairs. and you can access them by an index of another type.

Example:

```
Import java.util.Map;
Import java.util.HashMap;

class Main{
    Public static void Main (String [] args) {
Map<Integer, String> fruits = new HashMap<>();
        fruits.Put (1, "apple");
        fruits.Put (2, "Orange");
        fruits.Put (3, "Mango");
    System.out.Println ("Map:" + fruit);
    }
}
```

Out Put

Map: {apple, Orange, Mango}