

# Practice Program: 05-08-2024

---

1. Create and Implement the java program by using map.

Program:

```
import java.util.HashMap;

import java.util.Map;

public class MapExample {

    public static void main(String[] args) {

        Map<String, Integer> studentGrades = new HashMap<>();

        studentGrades.put("John", 85);

        studentGrades.put("Alice", 92);

        studentGrades.put("Bob", 78);

        studentGrades.put("Charlie", 95);

        System.out.println("Student Grades:");

        printMap(studentGrades);

        System.out.println("\nGet John's Grade:");

        System.out.println(studentGrades.get("John"));

        studentGrades.put("John", 90);

        System.out.println("\nUpdate John's Grade:");

        System.out.println(studentGrades.get("John"));

        studentGrades.remove("Bob");

        System.out.println("\nRemove Bob's Grade:");

        printMap(studentGrades);

        System.out.println("\nContains Key 'Alice':");

        System.out.println(studentGrades.containsKey("Alice"));

        System.out.println("\nContains Value 95:");

        System.out.println(studentGrades.containsValue(95));
```

```

System.out.println("\nMap Size:");

System.out.println(studentGrades.size());

studentGrades.clear();

System.out.println("\nClear Map:");

printMap(studentGrades);
}

public static void printMap(Map<String, Integer> map) {

    for (Map.Entry<String, Integer> entry : map.entrySet()) {

        System.out.println(entry.getKey() + ": " + entry.getValue());

    }

}

}

```

The screenshot shows the Programiz Online Java Compiler interface. The browser tabs include 'Overview | Oracle Academy', 'Oracle Academy', 'create and Implement the java', '(43) WhatsApp', and 'Online Java Compiler - Programiz'. The address bar shows 'programiz.com/java-programming/online-compiler/'.

The main editor displays a file named 'Main.java' with the following code:

```

1- import java.util.HashMap;
2- import java.util.Map;
3- public class MapExample {
4-     public static void main(String[] args) {
5-         Map<String, Integer> studentGrades = new HashMap<>();
6-         studentGrades.put("John", 85);
7-         studentGrades.put("Alice", 92);
8-         studentGrades.put("Bob", 78);
9-         studentGrades.put("Charlie", 95);
10-        System.out.println("Student Grades:");
11-        printMap(studentGrades);
12-        System.out.println("\nGet John's Grade:");
13-        System.out.println(studentGrades.get("John"));
14-        studentGrades.put("John", 90);
15-        System.out.println("\nUpdate John's Grade:");
16-        System.out.println(studentGrades.get("John"));
17-        studentGrades.remove("Bob");
18-        System.out.println("\nRemove Bob's Grade:");
19-        printMap(studentGrades);
20-        System.out.println("\nContains Key 'Alice':");
21-        System.out.println(studentGrades.containsKey("Alice"));
22-        System.out.println("\nContains Value 95:");

```

The output window on the right shows the following results:

```

java -cp /tmp/qRG84vKpho/MapExample
Student Grades:
Bob: 78
Alice: 92
Charlie: 95
John: 85

Get John's Grade:
85

Update John's Grade:
90

Remove Bob's Grade:
Alice: 92
Charlie: 95
John: 90

Contains Key 'Alice':
true

Contains Value 95:

```

The bottom of the image shows a Windows taskbar with the search bar and several application icons. The system clock indicates 13:16 on 05-08-2024.

2. Create a program using linked list for stack.

Program:

```
import java.util.LinkedList;

public class Stack{

    public static void main(String args[]){

        LinkedList<String> letter=new LinkedList<String>();

        letter.push("apple");

        letter.push("ball");

        letter.push("cat");

        letter.push("dog");

        System.out.println("Linked list : "+letter);

        System.out.println("Stack Size: " +letter.size());

        if(!letter.isEmpty()){

            System.out.println("poped element: "+letter.pop());

        }

    }

}
```

The screenshot shows a web browser with multiple tabs, including 'Online Java Compiler'. The active tab displays the Programiz Online Java Compiler interface. The main editor shows a Java file named 'Main.java' with the following code:

```
1- import java.util.LinkedList;
2- public class Stack{
3-     public static void main(String args[]){
4-         LinkedList<String> letter=new LinkedList<String>();
5-         letter.push("apple");
6-         letter.push("ball");
7-         letter.push("cat");
8-         letter.push("dog");
9-         System.out.println("Linked list : "+letter);
10        System.out.println("Stack Size: " +letter.size());
11        if(!letter.isEmpty()){
12            System.out.println("poped element: "+letter.pop());
13        }
14    }
15 }
```

The 'Output' panel on the right shows the results of the code execution:

```
java -cp /tmp/vDFfOUNdZH/Stack
Linked list : [dog, cat, ball, apple]
Stack Size: 4
poped element: dog
=== Code Execution Successful ===
```

The bottom of the image shows a Windows taskbar with the search bar and various application icons, including the Start menu, File Explorer, and web browsers.

3. Create a program using linked list for queue.

Program:

```
import java.util.LinkedList;

public class Queue{

    public static void main(String args[]){

        LinkedList<String> letter=new LinkedList<String>();

        letter.add("a");

        letter.add("b");

        letter.add("c");

        letter.add("d");

        System.out.println("Linked list : "+letter);

        System.out.println("Queue Size: "+letter.size());

        if(!letter.isEmpty()){

            System.out.println(letter.removeFirst());

        }


        System.out.println("Linked list : "+letter);

    }

}
```

Overview | Oracle Academy | (43) WhatsApp | Online Java Compiler | Online Java Compiler | Online Java Compiler | Online Java Compiler


programiz.com/java-programming/online-compiler/

  
Online Java Compiler

LEARN PYTHON

LOOKING TO LEARN PROGRAMMING?

Start your programming journey with Programiz AT NO COST.



Programiz PRO >

Main.java

1 import java.util.LinkedList;

2 public class Queue{

3 public static void main(String args[]){

4 LinkedList<String> letter=new LinkedList<String>();

5 letter.add("ant");

6 letter.add("bat");

7 letter.add("cup");

8 letter.add("duck");

9 System.out.println("Linked list : "+letter);

10 System.out.println("Queue Size: "+letter.size());

11 if(!letter.isEmpty()){

12 System.out.println(letter.removeFirst());

13 }

14 System.out.println("Linked list : "+letter);

15 }

16 }

Run

Share

Clear

Output

Clear


java -cp /tmp/g6IxxDlfih/Queue  
Linked list : [ant, bat, cup, duck]  
Queue Size: 4  
ant  
Linked list : [bat, cup, duck]  
  
=== Code Execution Successful ===

JS

GO

php

Type here to search



ENG

13:56

05-08-2024

4. Create mobile class with field price brand and so on take any one field and sort the class

Program:

```
import java.util.Arrays;

import java.util.Comparator;

class Mobile {

    private double price;

    private String brand;

    private String model;

    private int ram;

    public Mobile(double price, String brand, String model, int ram) {

        this.price = price;

        this.brand = brand;

        this.model = model;

        this.ram = ram;

    }

    public double getPrice() {

        return price;

    }

    public String getBrand() {

        return brand;

    }

    public String getModel() {

        return model;

    }

    public int getRam() {

        return ram;

    }

}
```

```

public String toString() {
    return "Mobile{" +
        "price=" + price +
        ", brand='" + brand + '\'' +
        ", model='" + model + '\'' +
        ", ram=" + ram +
        '}';
}

}

public class MobileSorter {

    public static void main(String[] args) {
        Mobile[] mobiles = new Mobile[] {
            new Mobile(15000, "Samsung", "Galaxy M31", 6),
            new Mobile(20000, "Apple", "iPhone 12", 4),
            new Mobile(10000, "Xiaomi", "Redmi Note 9", 4),
            new Mobile(25000, "OnePlus", "Nord", 8),
            new Mobile(12000, "Realme", "6 Pro", 6)
        };

        System.out.println("Before sorting:");

        Arrays.stream(mobiles).forEach(System.out::println);

        Arrays.sort(mobiles, Comparator.comparingDouble(Mobile::getPrice));

        System.out.println("\nAfter sorting by price:");

        Arrays.stream(mobiles).forEach(System.out::println);
    }

}

```

Overview | Oracle Academy | create and Implement | (43) WhatsApp | Online Java Compiler | Online Java Compiler

programiz.com/java-programming/online-compiler/

Programiz

Online Java Compiler

LEARN PYTHON

LOOKING TO LEARN PROGRAMMING?

Start your programming journey with Programiz AT NO COST.

Programiz PRO

Main.java

Run

Share

Clear

```
1 import java.util.Arrays;
2 import java.util.Comparator;
3 class Mobile {
4     private double price;
5     private String brand;
6     private String model;
7     private int ram;
8     public Mobile(double price, String brand, String model, int ram)
9     {
10         this.price = price;
11         this.brand = brand;
12         this.model = model;
13         this.ram = ram;
14     }
15     public double getPrice() {
16         return price;
17     }
18     public String getBrand() {
19         return brand;
20     }
21     public String getModel() {
22         return model;
23     }
24 }
```

```
java -cp /tmp/jrC6Kk0hE8/MobileSorter
Before sorting:
Mobile{price=15000.0, brand='Samsung', model='Galaxy M31', ram=6}
Mobile{price=20000.0, brand='Apple', model='iPhone 12', ram=4}
Mobile{price=10000.0, brand='Xiaomi', model='Redmi Note 9', ram=4}
Mobile{price=25000.0, brand='OnePlus', model='Nord', ram=8}
Mobile{price=12000.0, brand='Realme', model='6 Pro', ram=6}

After sorting by price:
Mobile{price=10000.0, brand='Xiaomi', model='Redmi Note 9', ram=4}
Mobile{price=12000.0, brand='Realme', model='6 Pro', ram=6}
Mobile{price=15000.0, brand='Samsung', model='Galaxy M31', ram=6}
Mobile{price=20000.0, brand='Apple', model='iPhone 12', ram=4}
Mobile{price=25000.0, brand='OnePlus', model='Nord', ram=8}

=== Code Execution Successful ===
```

Type here to search

13:49 05-08-2024