CSE 1007 LAB 9

-AISHWARYA S 19BCE1709

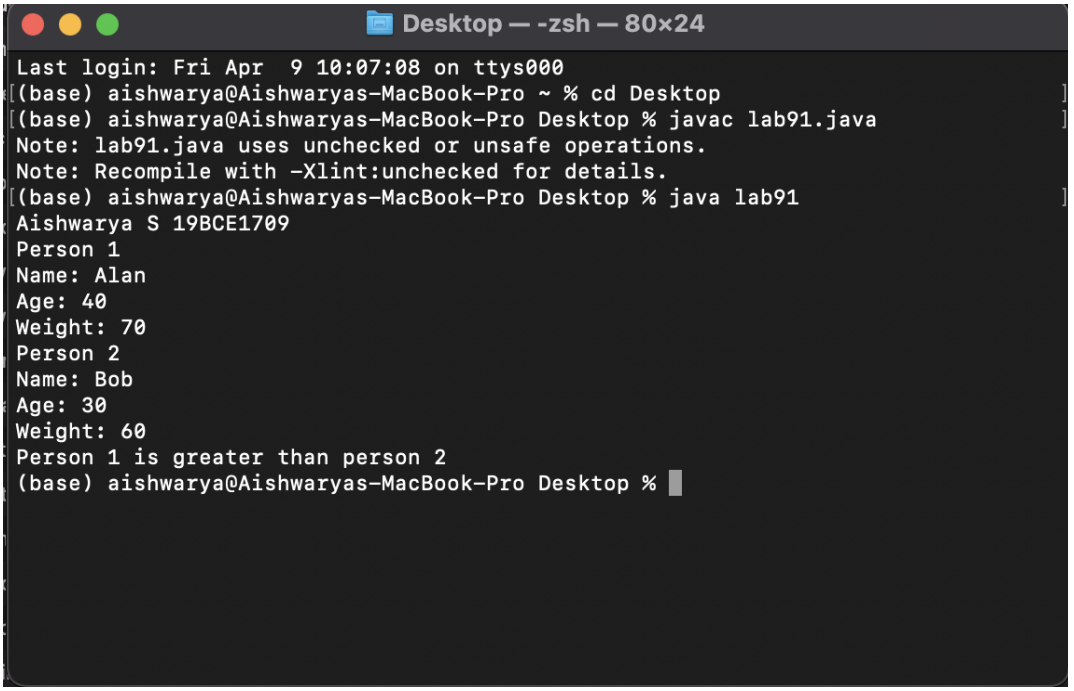
1. Lambda function

CODE:

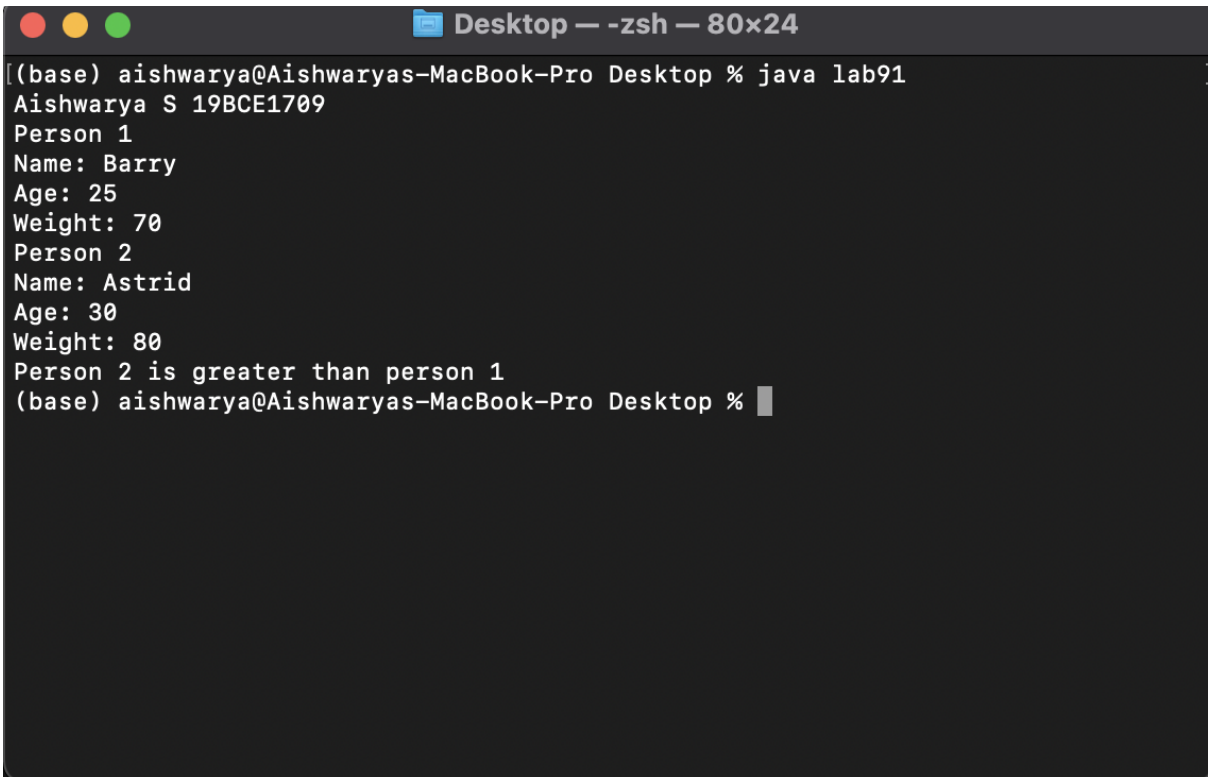
```
import java.util.*;
import java.lang.*;
class lab91
{
@FunctionalInterface
interface funcparam<p1, p2>
{
   public void param(p1 pone,p2 ptwo);

}
public static void main(String args[])
{    System.out.print("Aishwarya S 19BCE1709\n");
   Map<Object,ArrayList<String>> personlist = new HashMap<>();
   String name, age, weight;
   ArrayList<String>[] arr = new ArrayList[2];
   Scanner s= new Scanner(System.in);
   for(int i=1;i<=2;i++)
     {    arr[i-1]  = new ArrayList<String>();
       System.out.print("Person "+ i+"\n");
         System.out.print("Name: ");
         name=s.nextLine();
          arr[i-1].add(name);


         System.out.print("Age: ");
         age=s.nextLine();
         arr[i-1].add(age);

          System.out.print("Weight: ");
          weight= s.nextLine();
          arr[i-1].add(weight);



     }
```

```
funcparam<ArrayList<String>, ArrayList<String>>  m = ( a, b) -> {
 if(a.get(0).compareTo(b.get(0))<0&&
Integer.parseInt(a.get(1))>Integer.parseInt(b.get(1)) &&
Float.parseFloat(a.get(2))>Float.parseFloat(b.get(2)))
      System.out.print("Person 1 is greater than person 2\n");
      else if(a.get(0).compareTo(b.get(0))>0&&
Integer.parseInt(a.get(1))<Integer.parseInt(b.get(1)) &&
Float.parseFloat(a.get(2))<Float.parseFloat(b.get(2)))
        System.out.print("Person 2 is greater than person 1\n");
        else
          System.out.print("Neither person is greater thean the other\n");

};


 m.param(arr[0],arr[1]);



}
}
```

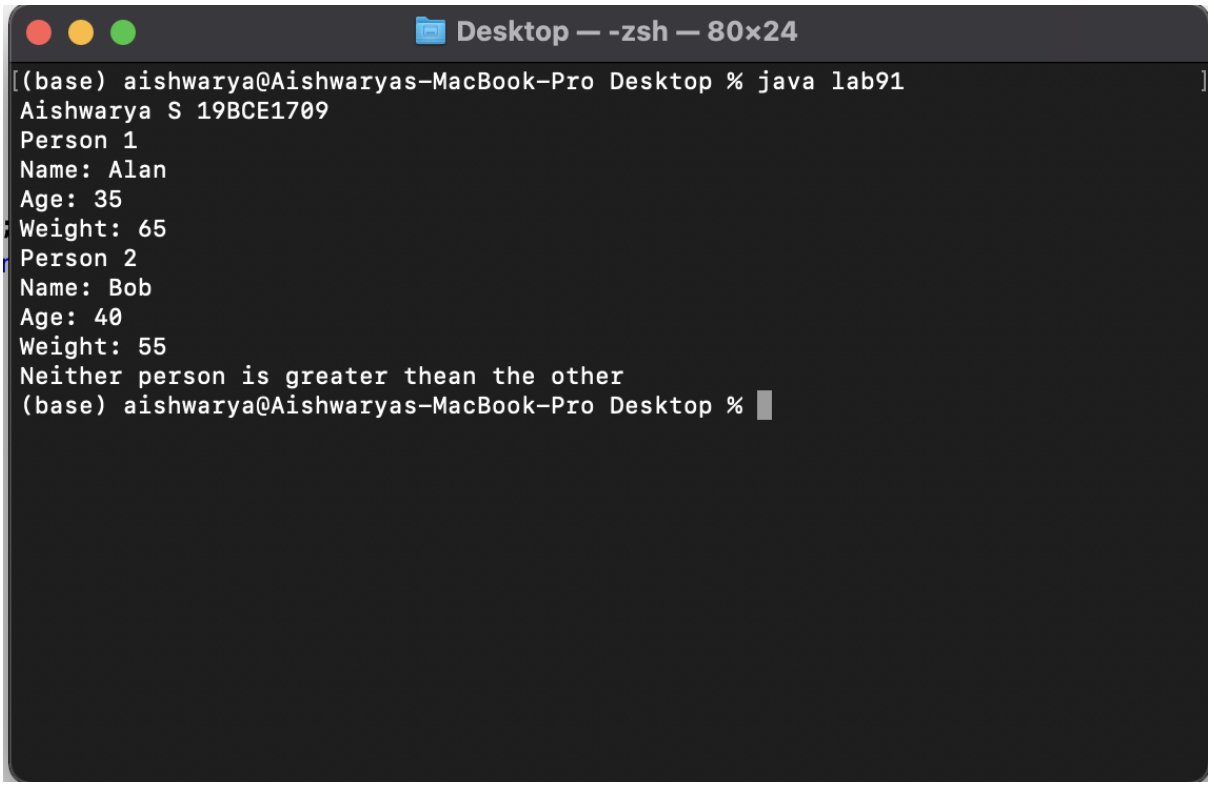
TEST CASE 1:



TEST CASE 2:



TEST CASE 3:



2. - Add or remove a student from h1

- Iterate over the maps and display the key-value pairs stored in them
- Given a student name, fetch the names of all those who teach him/her.

CODE:

```
import java.util.*;
import java.lang.*;
class java92
{
   static void removestu(Map<String,ArrayList<String>> students)
 {  System.out.print("\nEnter the name of the student to be removed\n");
   System.out.println(students+"\n");

   Scanner s= new Scanner(System.in);
    String st= s.nextLine();
      students.remove(st);
    System.out.print("\nStudent has been removed\n");



 }
 static void addstu(Map<String,ArrayList<String>> students)
   {  System.out.println("\n"+students);
     Scanner s= new Scanner(System.in);
     ArrayList<String> subjs= new ArrayList();
     System.out.print("\nEnter the details of the student to be added\n");
     System.out.print("\nName: ");
           String name=s.nextLine();
           System.out.print("Enter the subjects: ");
          String m=s.nextLine();
           String sub[]=m.split("\\s+");
           while(sub.length>3)
             {
                  System.out.print("Max 3; enter again:  ");
                   m=s.nextLine();
                  sub=m.split("\\s+");

             }
           for(String str: sub)
           { subjs.add(str);

           }
           students.put(name,subjs);


   }
```

```
 static void dispmaps(Map<String,ArrayList<String>> m1)
 {
     for (Map.Entry<String, ArrayList<String>> entry : m1.entrySet())
               {    System.out.print("\nName:  "+  entry.getKey()+"\nSubjects:
"+entry.getValue());
       }
 }
          static          void          findf(Map<String,ArrayList<String>>
m1,Map<String,ArrayList<String>> m2 ) throws NullPointerException
 {  HashSet<String> teachers=new HashSet();
      System.out.print("\n\nEnter the name of the student whose faculties you
want to find: ");
  Scanner s= new Scanner(System.in);
    String st= s.nextLine();
     ArrayList<String> sub_l =sub_l= m1.get(st);




     try{
   for(String ss: sub_l)
      {
          for(Map.Entry<String, ArrayList<String>> entry : m2.entrySet())
            {
                if(entry.getValue().contains(ss))
                    teachers.add(entry.getKey());



      }




 }
     }
      catch(Exception n)
     {
         System.out.print("Student does not exist\n");
     }
      System.out.print("The faculties are: ");
     for(String hs:teachers )
       {
           System.out.print(hs+" ");
       }
```

```
      System.out.print("\n");
}
  public static void main(String args[]) throws NullPointerException
    {   Map<String,ArrayList<String>> students = new HashMap<>();
     Map<String,ArrayList<String>> teacher_course = new HashMap<>();
         ArrayList<String> subjs= new ArrayList();

        System.out.print("AISHWARYA S 19BCE1709\n");
        Scanner s= new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n=s.nextInt();
        s.nextLine();
        String m;
        String name;
        for(int i=0;i<n;i++)
          { subjs= new ArrayList();

              boolean f=false;
              System.out.print("\nName: ");
          name=s.nextLine();
          System.out.print("Enter the subjects: ");
          m=s.nextLine();
          String sub[]=m.split("\\s+");
          while(sub.length>3)
            {
                  System.out.print("Max 3; enter again:  ");
                   m=s.nextLine();
                  sub=m.split("\\s+");

            }
          for(String str: sub)
          { subjs.add(str);

          }
          students.put(name,subjs);




          }

          System.out.print("\nEnter the number of faculties: ");
          int n1=s.nextInt();
           s.nextLine();
           for(int i=0;i<n1;i++)
```

```
            {   subjs= new ArrayList();
                boolean f=false;
                System.out.print("\nName: ");
             name=s.nextLine();
             System.out.print("Enter the subjects: ");
             m=s.nextLine();
             String sub[]=m.split("\\s+");

             for(String str: sub)
             { subjs.add(str);

             }
            teacher_course.put(name,subjs);




             }
          removestu(students);
          addstu(students);
          System.out.print("\nStudent List\n");
          dispmaps(students);
          System.out.print("\n\nFaculty List\n");
          dispmaps(teacher_course);
          findf(students, teacher_course);


      }



}
```

OUTPUT:

```
(base) aishwarya@Aishwaryas-MacBook-Pro Desktop % javac java92.java
Note: java92.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
(base) aishwarya@Aishwaryas-MacBook-Pro Desktop % java java92
AISHWARYA S 19BCE1709
Enter the number of students: 3

Name: Arthur
Enter the subjects: STS English C++ Maths
Max 3; enter again:  STS English C++

Name: Sadie
Enter the subjects: STS C++

Name: Marston
Enter the subjects: English

Enter the number of faculties: 3

Name: Dutch
Enter the subjects: Java C++

Name: Hosea
Enter the subjects: Maths

Name: Pearson
Enter the subjects: STS English

Enter the name of the student to be removed
{Arthur=[STS, English, C++], Marston=[English], Sadie=[STS, C++]}

Sadie

Student has been removed

{Arthur=[STS, English, C++], Marston=[English]}

Enter the details of the student to be added

Name: Molly
Enter the subjects: Java STS

Student List

Name: Arthur
Subjects: [STS, English, C++]
Name: Molly
Subjects: [Java, STS]
Name: Marston
Subjects: [English]

Faculty List

Name: Hosea
Subjects: [Maths]
Name: Pearson
Subjects: [STS, English]
Name: Dutch
Subjects: [Java, C++]

Enter the name of the student whose faculties you want to find: Molly
The faculties are: Pearson Dutch
(base) aishwarya@Aishwaryas-MacBook-Pro Desktop %
```

3. Identify the students who had took all breakfast, lunch, dinner and none of these in the mess

CODE:

```
import java.util.*;
class java93<T>
{
  Set<T> s = new HashSet<T>();
  Set<T> b = new HashSet<T>();
Set<T> l = new HashSet<T>();
   Set<T> d = new HashSet<T>();

  public void addtostu(T t)

   {
       s.add(t);

   }

public void addtob(T t)

   {
       b.add(t);
   }
   public void addtol(T t)

   {
       l.add(t);
   }
   public void addtod(T t)

   {
       d.add(t);
   }
   public  void find_a()
     {   Set<T> as = new HashSet<T>(s);

         System.out.print("Students who took breakfast, dinner, and
lunch\n");
         as.retainAll(b);
         as.retainAll(l);
         as.retainAll(d);
         System.out.println(as);
     }
     public void find_n()
     {   Set<T> ns = new HashSet<T>(s);
```

```
System.out.print("Students who did not take anything\n");
         ns.removeAll(b);
          ns.removeAll(l);
           ns.removeAll(d);


         System.out.println(ns);

     }
public static  void main(String[] args)
{   java93 obj= new java93();
   System.out.print("Aishwarya S 19BCE1709\n");
Scanner sc = new Scanner(System.in);
String c;
int ch;
System.out.print("Enter the data type you want to use\n(String 1, Character
2,Integer: 3,  Float 4): ");
ch=sc.nextInt();
sc.nextLine();
//String[] arr=new String[] {"Integer", "String","Character","Double",
"Long"};

System.out.print("Enter the students: ");
c= sc.nextLine();
String arr[]= c.split("\\s+");
for(String st: arr)
  {
     if(ch==1)
        obj.addtostu(st);
       else if(ch==2)
          obj.addtostu(st.charAt(0));
       else if(ch==3)
          obj.addtostu(Integer.parseInt(st));
       else
           obj.addtostu(Float.parseFloat(st));
  }


System.out.print("Enter the students who took breakfast: ");
c= sc.nextLine();

arr= c.split("\\s+");

for(String st: arr)
```

```
    { if(ch==1)
        obj.addtob(st);
       else if(ch==2)
          obj.addtob(st.charAt(0));
       else if(ch==3)
          obj.addtob(Integer.parseInt(st));
       else
           obj.addtob(Float.parseFloat(st));
    }
```

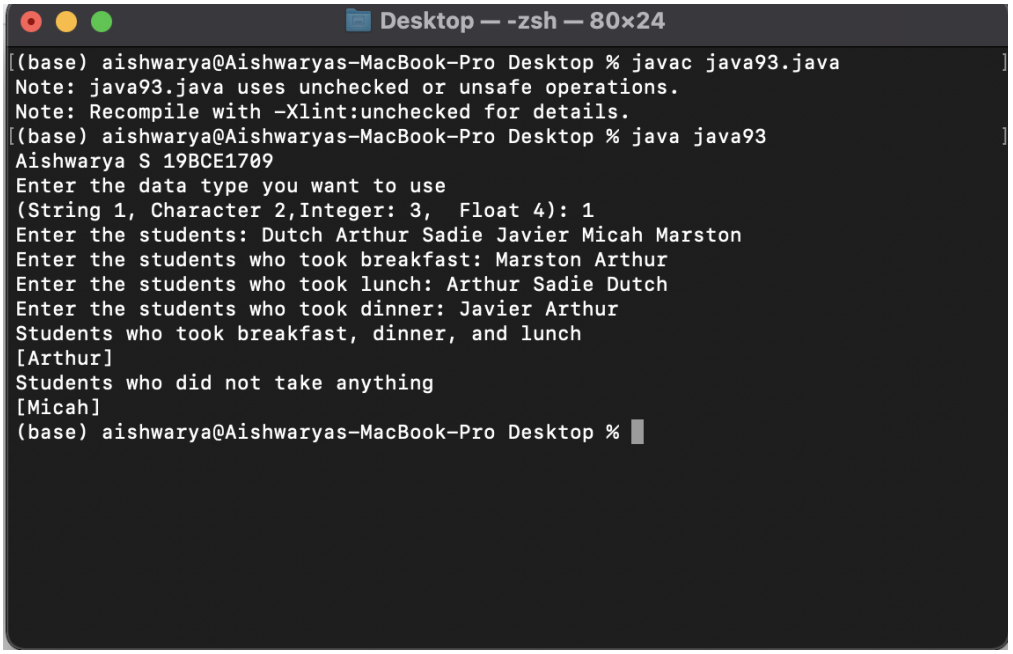
```
System.out.print("Enter the students who took lunch: ");
c= sc.nextLine();
arr= c.split("\\s+");
for(String st: arr)
{
     if(ch==1)
        obj.addtol(st);
       else if(ch==2)
          obj.addtol(st.charAt(0));
       else if(ch==3)
          obj.addtol(Integer.parseInt(st));
       else
           obj.addtol(Float.parseFloat(st));

 }
```

```
System.out.print("Enter the students who took dinner: ");
c= sc.nextLine();
arr= c.split("\\s+");
for(String st: arr)
{if(ch==1)
    obj.addtod(st);
       else if(ch==2)
          obj.addtod(st.charAt(0));
       else if(ch==3)
          obj.addtod(Integer.parseInt(st));
       else
           obj.addtod(Float.parseFloat(st));
}
```

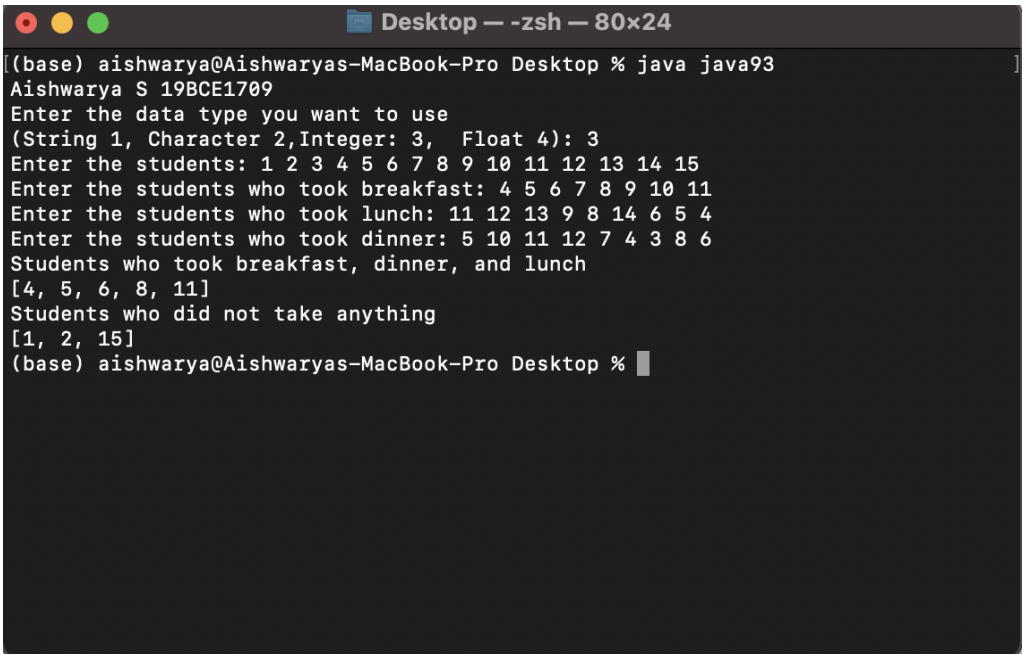
```
obj.find_a();
obj.find_n();


    }

}
```

OUTPUTS:

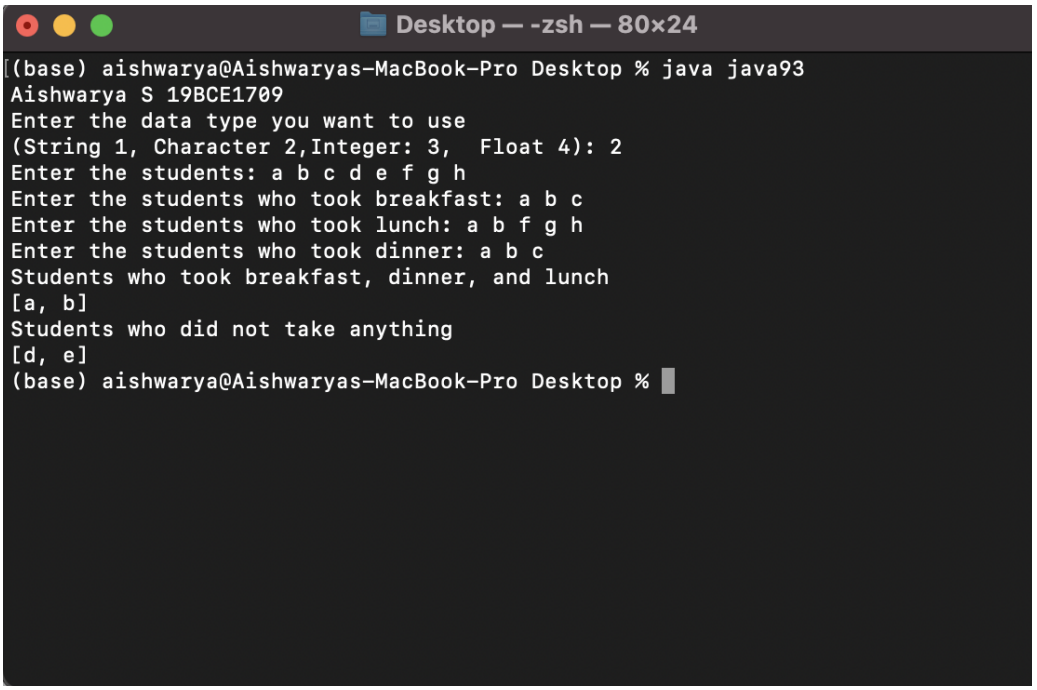
TEST CASE 1:



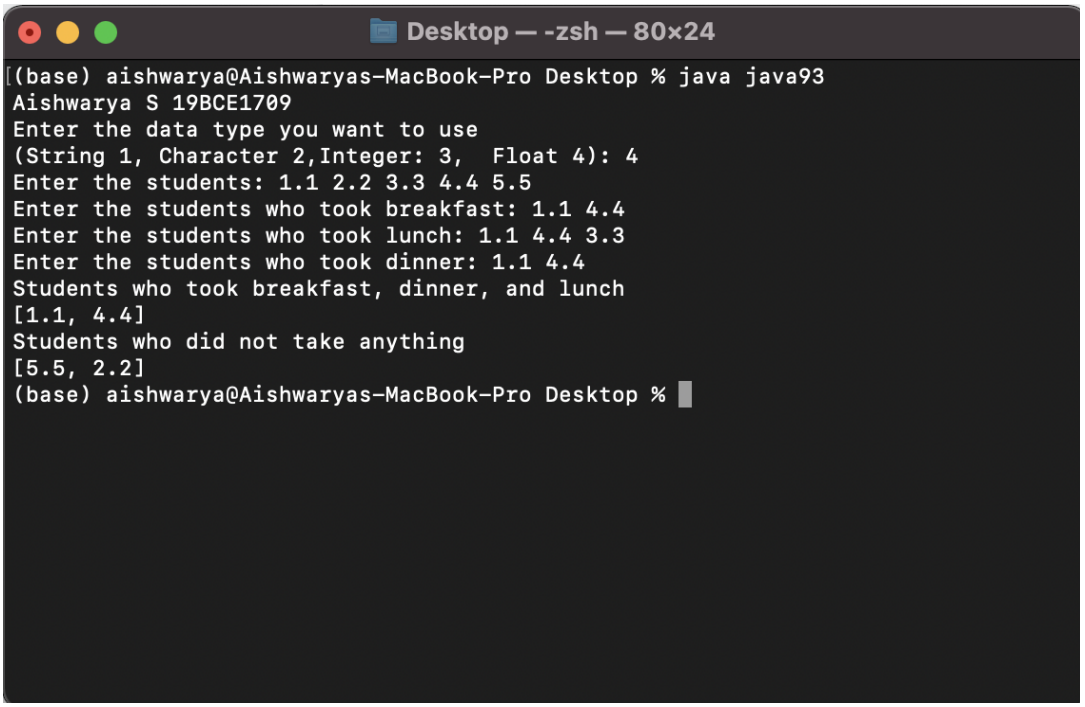
TEST CASE 2:



TEST CASE 3:



TEST CASE 4:


Desktop — -zsh — 80×24
(base) aishwarya@Aishwaryas-MacBook-Pro Desktop % java java93
Aishwarya S 19BCE1709
Enter the data type you want to use
(String 1, Character 2,Integer: 3, Float 4): 4
Enter the students: 1.1 2.2 3.3 4.4 5.5
Enter the students who took breakfast: 1.1 4.4
Enter the students who took lunch: 1.1 4.4 3.3
Enter the students who took dinner: 1.1 4.4
Students who took breakfast, dinner, and lunch
[1.1, 4.4]
Students who did not take anything
[5.5, 2.2]
(base) aishwarya@Aishwaryas-MacBook-Pro Desktop %