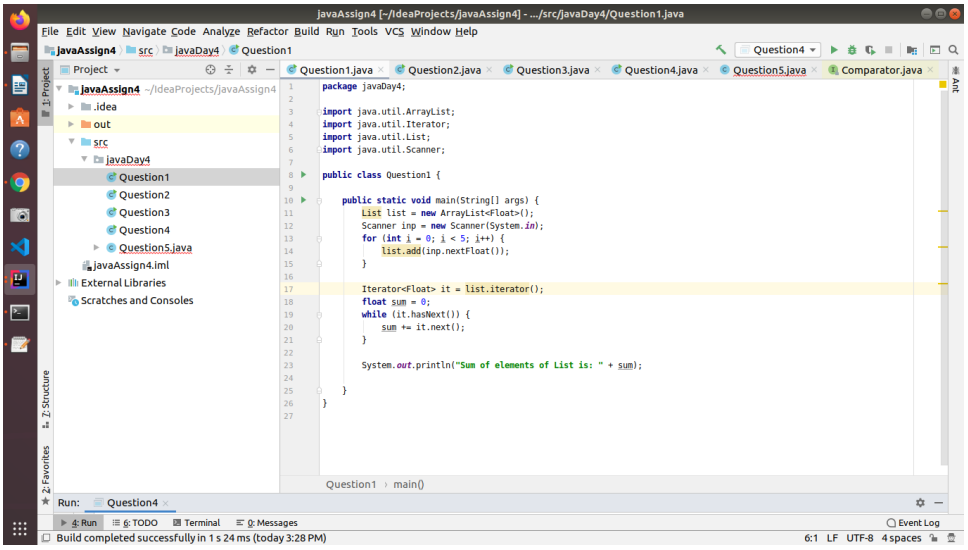
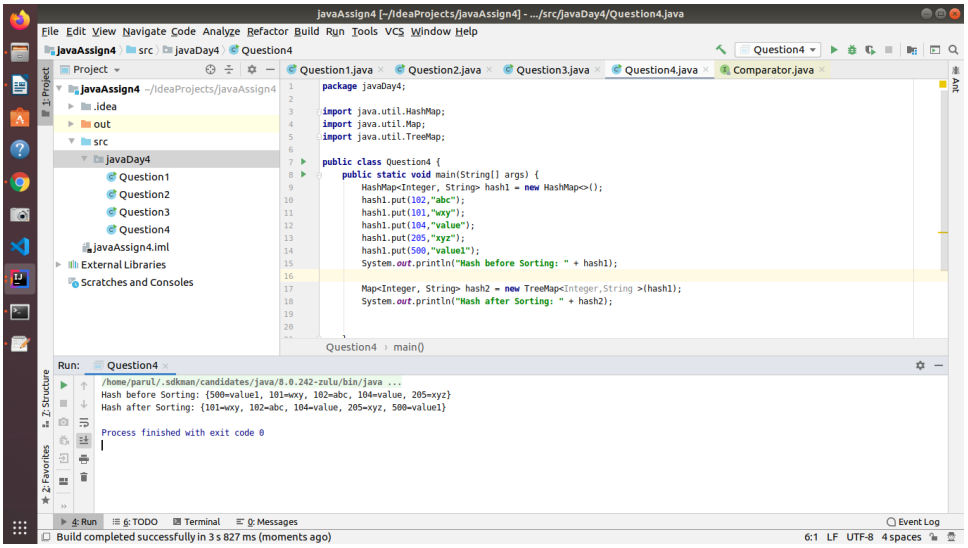
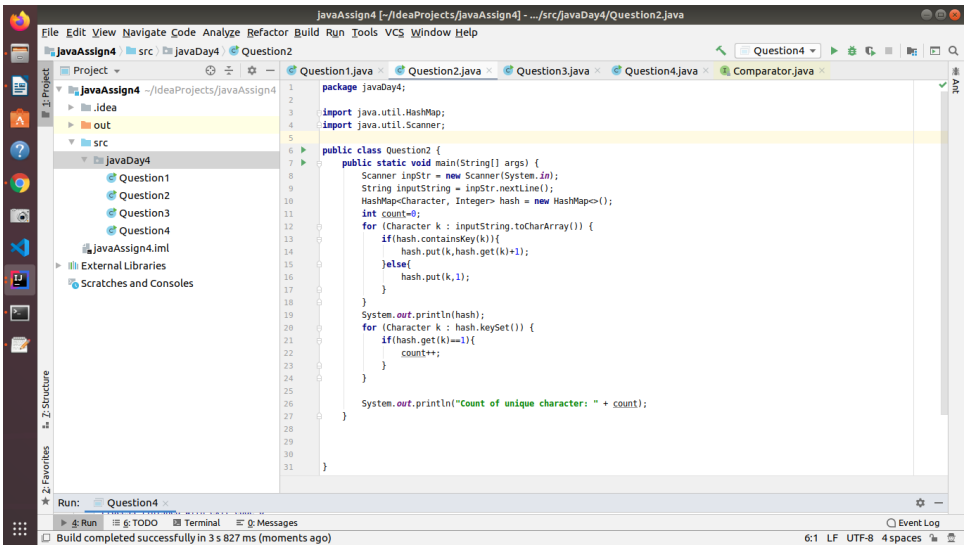
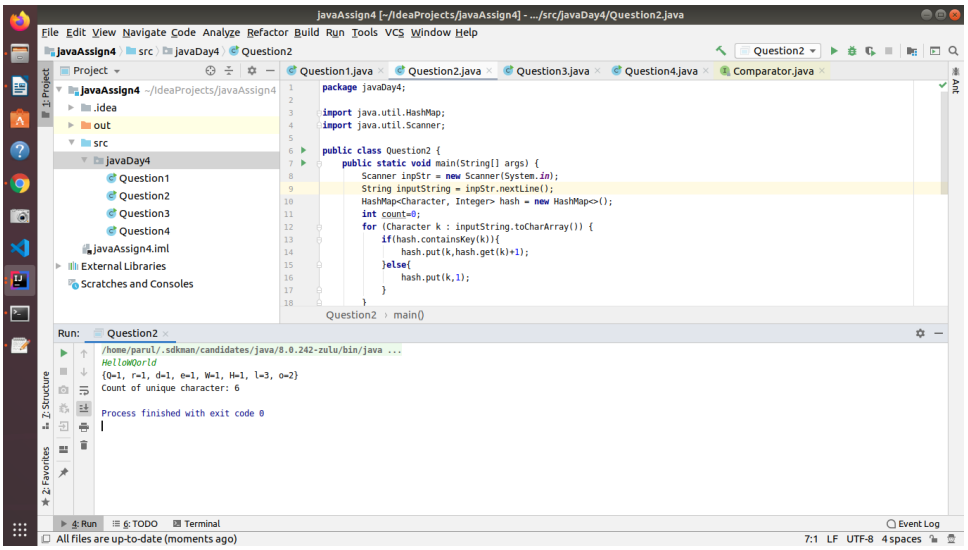
Question: 1:
Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.





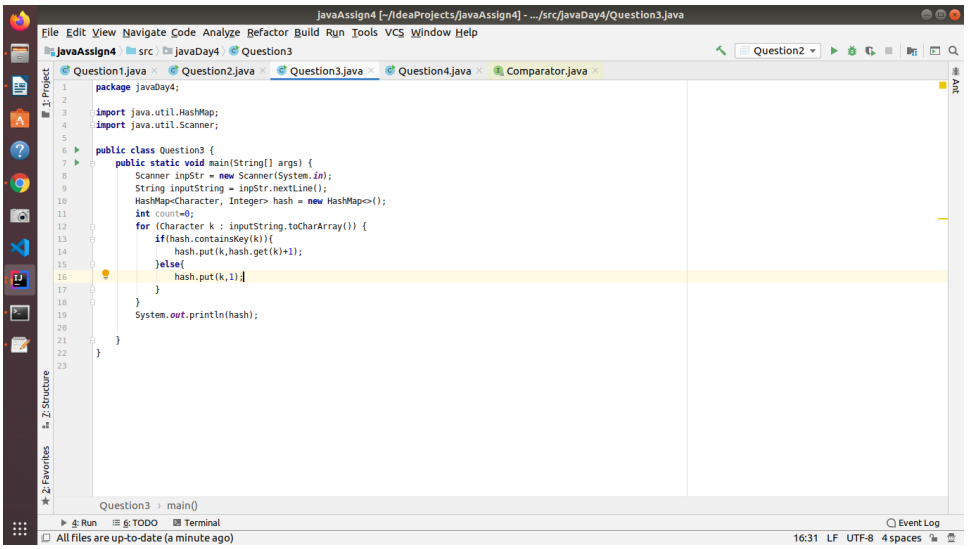
Question: 2:
Write a method that takes a string and returns the number of unique characters in the string.

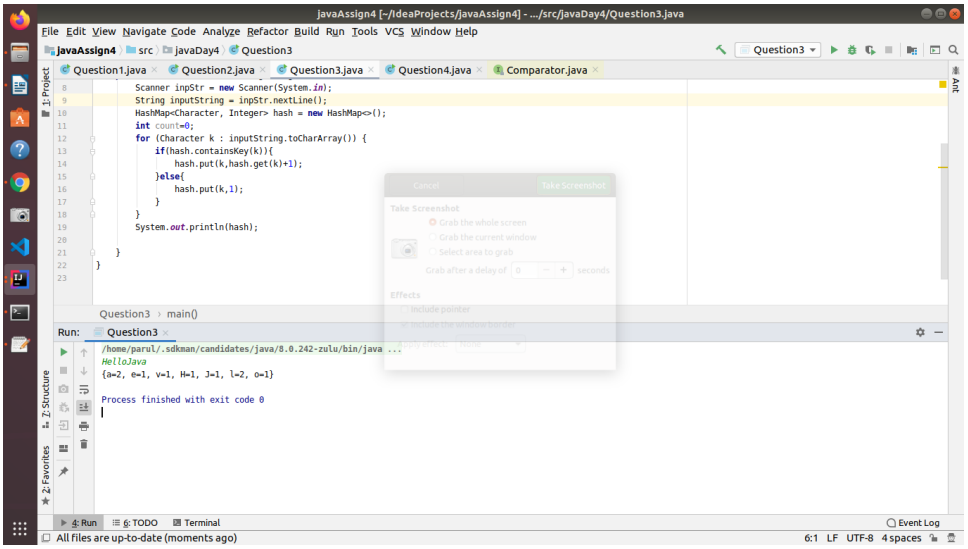




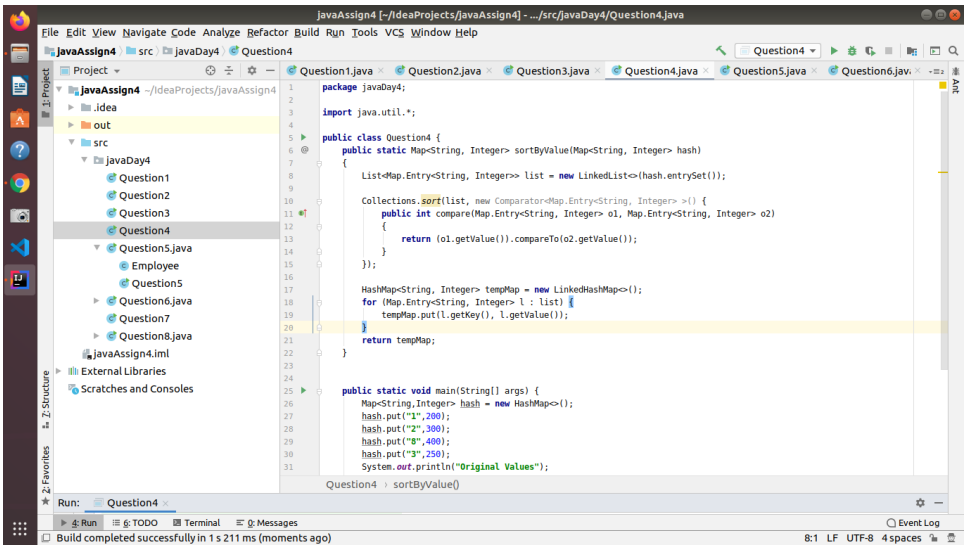
Question: 3:

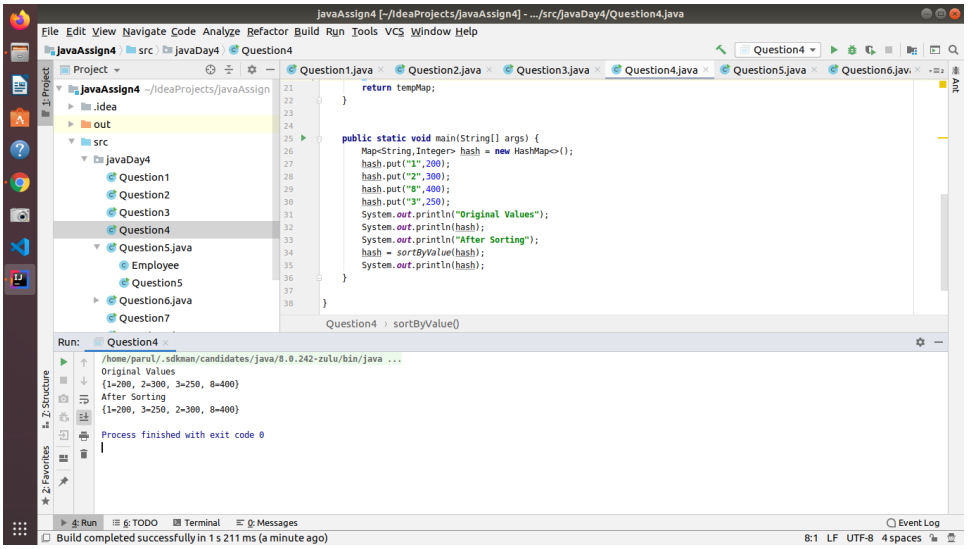
Write a method that takes a string and print the number of occurrence of each character characters in the string.





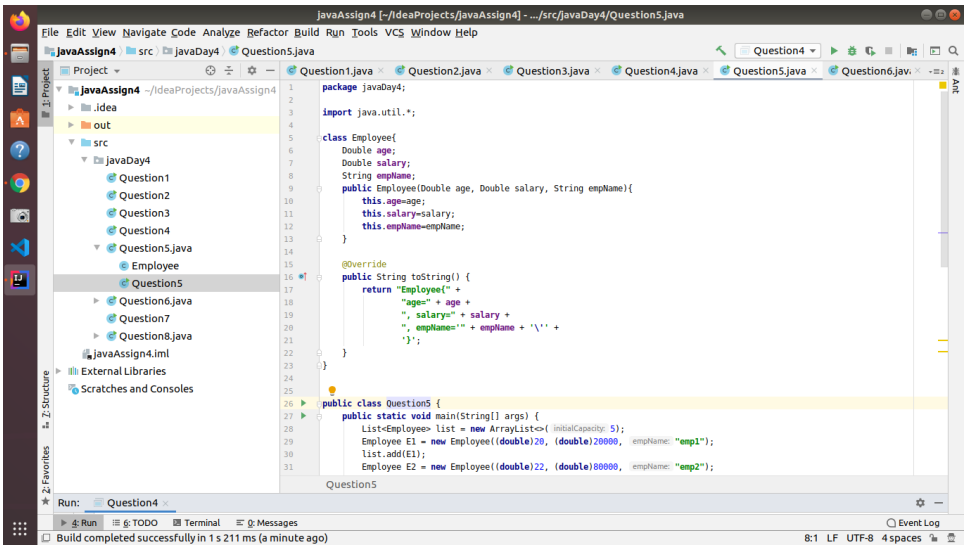
Question: 4:
Write a program to sort HashMap by value.

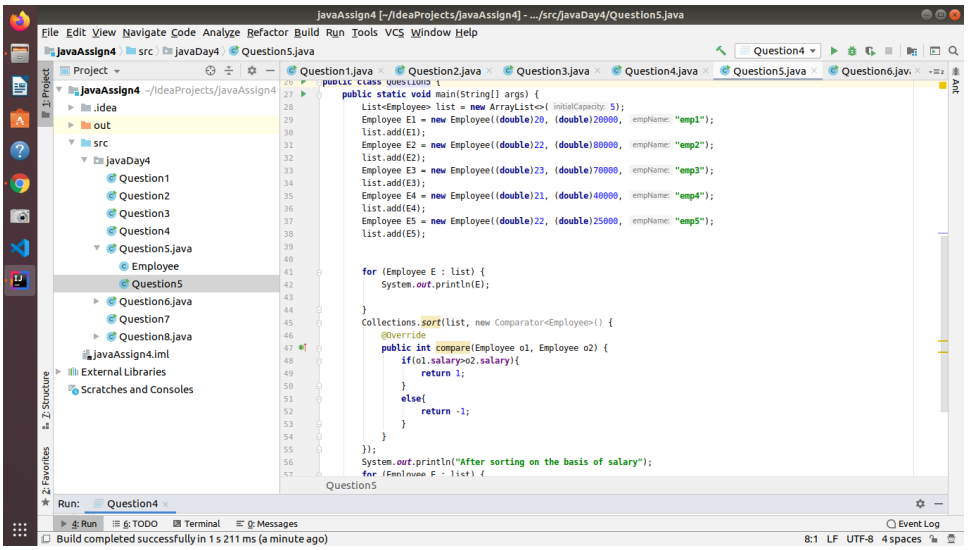


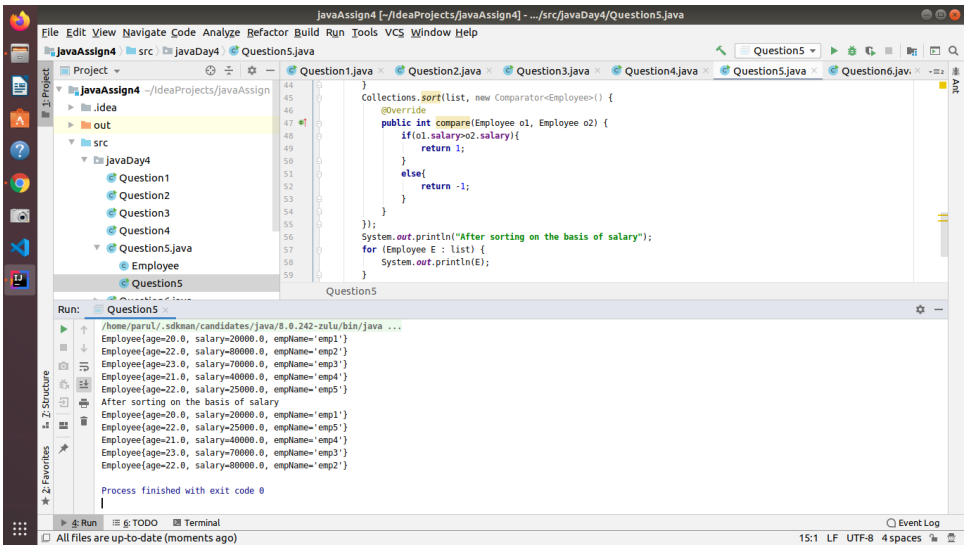


Question: 5:

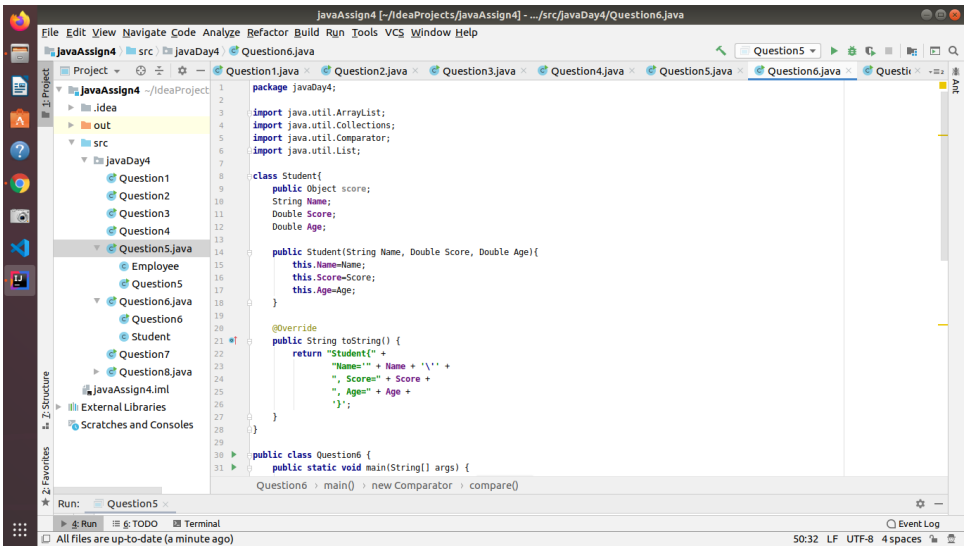
Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name

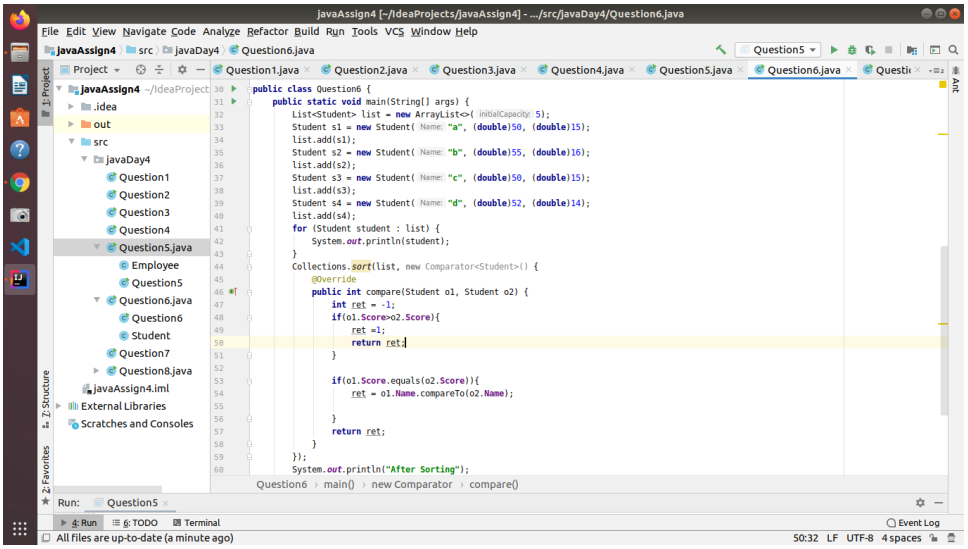

javaAssign4 [~/IdeaProjects/javaAssign4] - .../src/javaDay4/Question5.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaAssign4 › src › javaDay4 › Question5.java
Question4
Project
javaAssign4 ~/IdeaProjects/javaAssign4
.idea
out
src
javaDay4
Question1
Question2
Question3
Question4
Question5.java
Employee
Question5
Question6.java
Question7
Question8.java
javaAssign4.iml
External Libraries
Scratches and Consoles
Question1.java
Question2.java
Question3.java
Question4.java
Question5.java
Question6.java
package javaDay4;
import java.util.*;
class Employee{
Double age;
Double salary;
String empName;
public Employee(Double age, Double salary, String empName){
this.age=age;
this.salary=salary;
this.empName=empName;
}
@Override
public String toString() {
return "Employee{" +
"age=" + age +
", salary=" + salary +
", empName='" + empName + '\'' +
'}';
}
}
public class Question5 {
public static void main(String[] args) {
List<Employee> list = new ArrayList<>(initialCapacity: 5);
Employee E1 = new Employee((double)20, (double)20000, empName: "emp1");
list.add(E1);
Employee E2 = new Employee((double)22, (double)80000, empName: "emp2");
Question5
Run: Question4
4: Run
6: TODO
Terminal
0: Messages
Event Log
Build completed successfully in 1 s 211 ms (a minute ago)
8:1 LF UTF-8 4 spaces

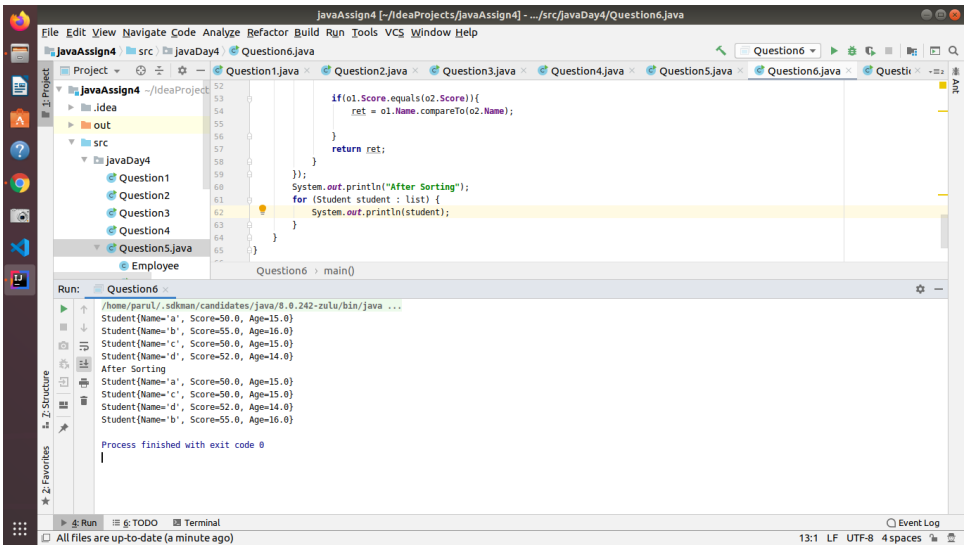

javaAssign4 [~/IdeaProjects/javaAssign4] - .../src/javaDay4/Question5.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaAssign4 › src › javaDay4 › Question5.java
Question4
Project
javaAssign4 ~/IdeaProjects/javaAssign4
.idea
out
src
javaDay4
Question1
Question2
Question3
Question4
Question5.java
Employee
Question5
Question6.java
Question7
Question8.java
javaAssign4.iml
External Libraries
Scratches and Consoles
Question1.java
Question2.java
Question3.java
Question4.java
Question5.java
Question6.java
public class Question5 {
public static void main(String[] args) {
List<Employee> list = new ArrayList<>(initialCapacity: 5);
Employee E1 = new Employee((double)20, (double)20000, empName: "emp1");
list.add(E1);
Employee E2 = new Employee((double)22, (double)80000, empName: "emp2");
list.add(E2);
Employee E3 = new Employee((double)23, (double)70000, empName: "emp3");
list.add(E3);
Employee E4 = new Employee((double)21, (double)40000, empName: "emp4");
list.add(E4);
Employee E5 = new Employee((double)22, (double)25000, empName: "emp5");
list.add(E5);
for (Employee E : list) {
System.out.println(E);
}
Collections.sort(list, new Comparator<Employee>() {
@Override
public int compare(Employee o1, Employee o2) {
if(o1.salary>o2.salary){
return 1;
}
else{
return -1;
}
}
});
System.out.println("After sorting on the basis of salary");
Question5
Run: Question4
4: Run
6: TODO
Terminal
0: Messages
Event Log
Build completed successfully in 1 s 211 ms (a minute ago)
8:1 LF UTF-8 4 spaces


javaAssign4 [~/IdeaProjects/javaAssign4] - .../src/javaDay4/Question5.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaAssign4 › src › javaDay4 › Question5.java
Question5
Project
javaAssign4 ~/IdeaProjects/javaAssign
.idea
out
src
javaDay4
Question1
Question2
Question3
Question4
Question5.java
Employee
Question5
Question1.java
Question2.java
Question3.java
Question4.java
Question5.java
Question6.java
}
Collections.sort(list, new Comparator<Employee>() {
@Override
public int compare(Employee o1, Employee o2) {
if(o1.salary>o2.salary){
return 1;
}
else{
return -1;
}
}
});
System.out.println("After sorting on the basis of salary");
for (Employee E : list) {
System.out.println(E);
}
Question5
Run: Question5
/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Employee{age=20.0, salary=20000.0, empName='emp1'}
Employee{age=22.0, salary=80000.0, empName='emp2'}
Employee{age=23.0, salary=70000.0, empName='emp3'}
Employee{age=21.0, salary=40000.0, empName='emp4'}
Employee{age=22.0, salary=25000.0, empName='emp5'}
After sorting on the basis of salary
Employee{age=20.0, salary=20000.0, empName='emp1'}
Employee{age=22.0, salary=25000.0, empName='emp5'}
Employee{age=21.0, salary=40000.0, empName='emp4'}
Employee{age=23.0, salary=70000.0, empName='emp3'}
Employee{age=22.0, salary=80000.0, empName='emp2'}
Process finished with exit code 0
4: Run
6: TODO
Terminal
Event Log
All files are up-to-date (moments ago)
15:1 LF UTF-8 4 spaces

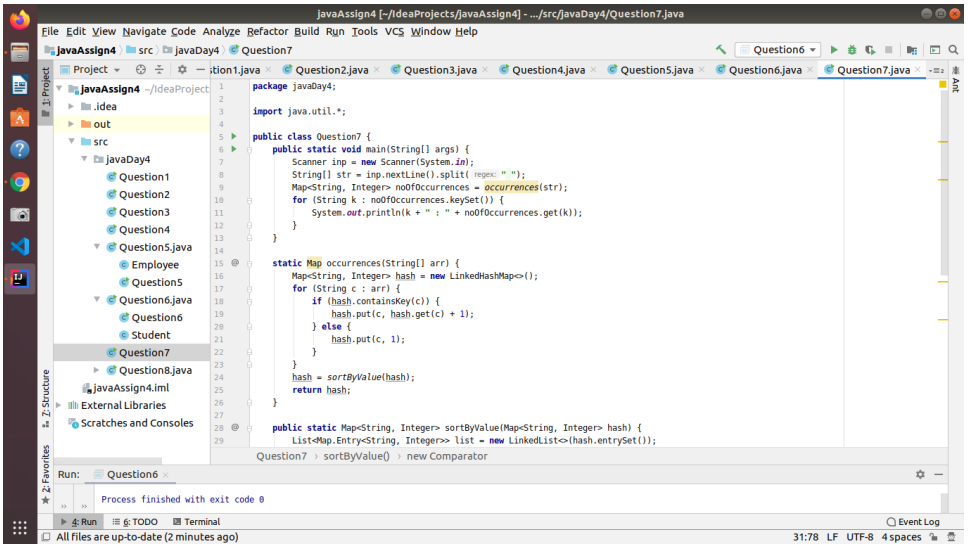
Question: 6:
Write a program to sort the Student objects based on Score , if the score are same then sort on First Name . Class Student{ String Name; Double Score; Double Age

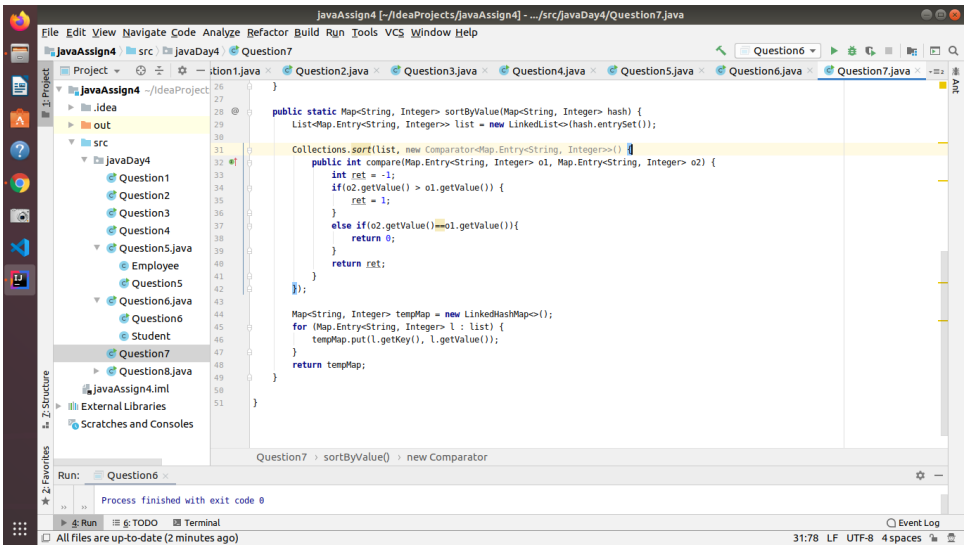


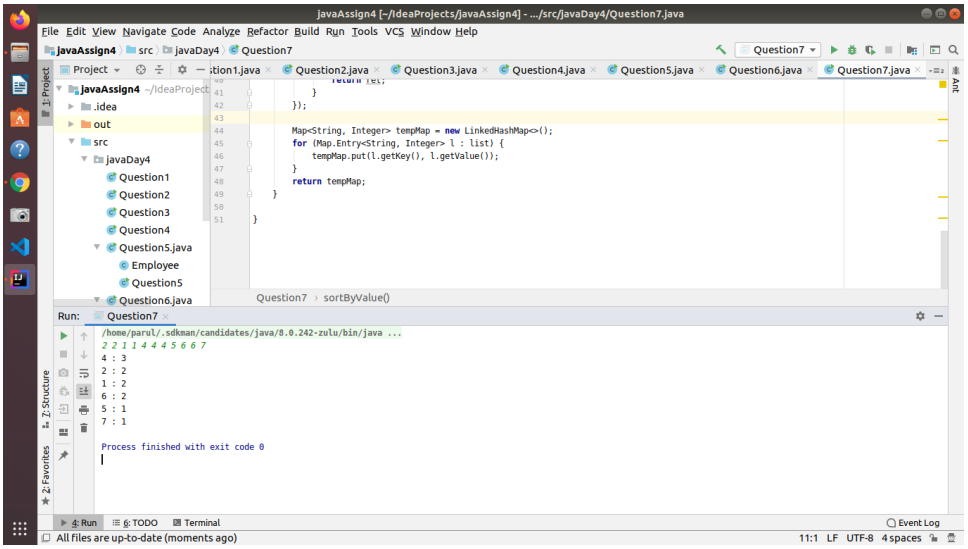




Question: 7:
Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.

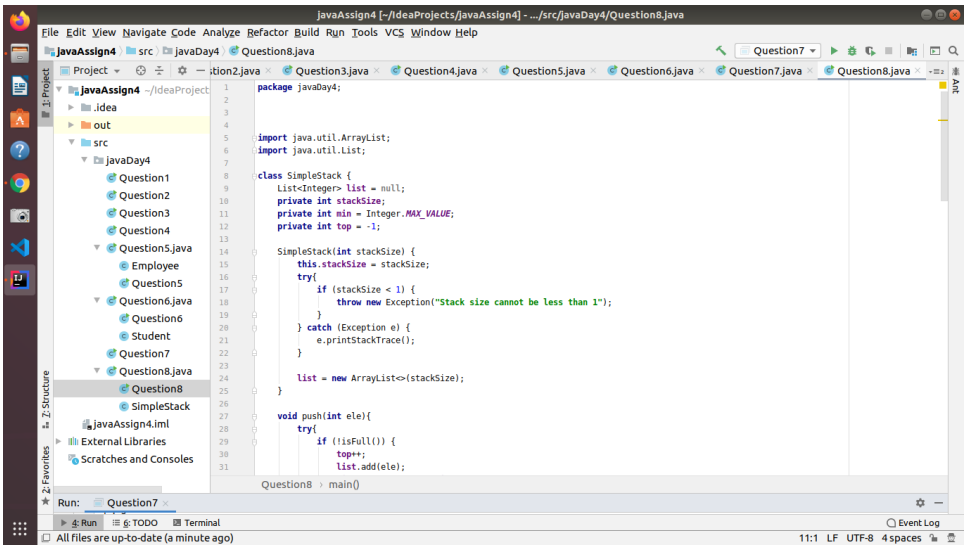


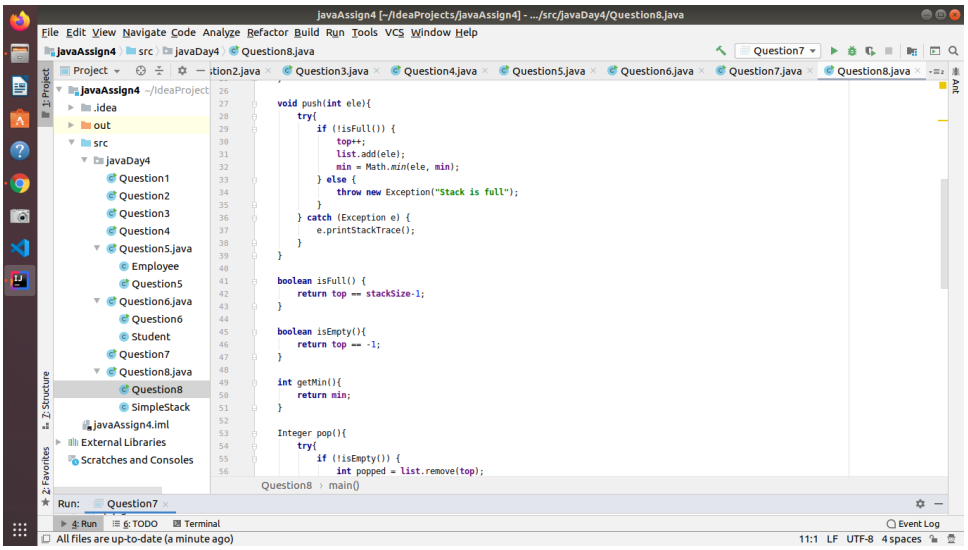


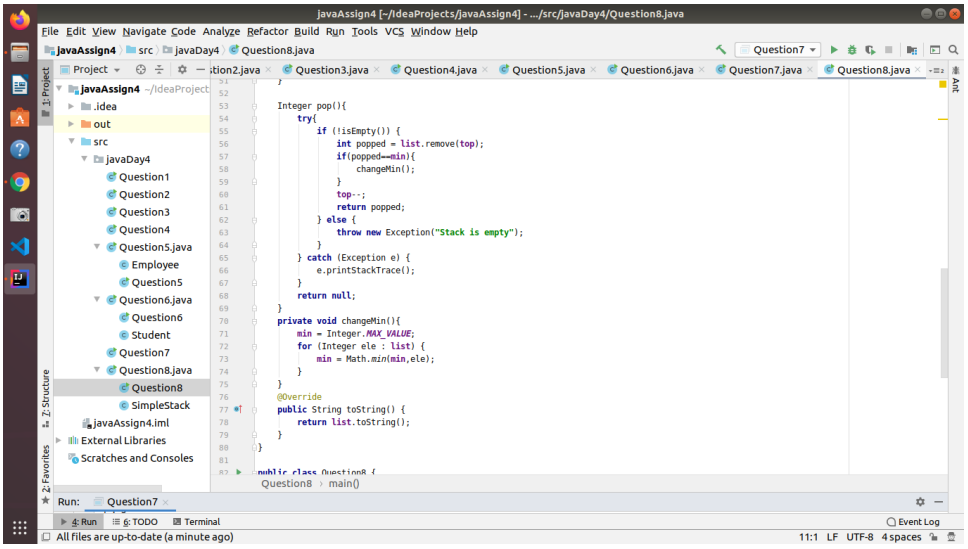


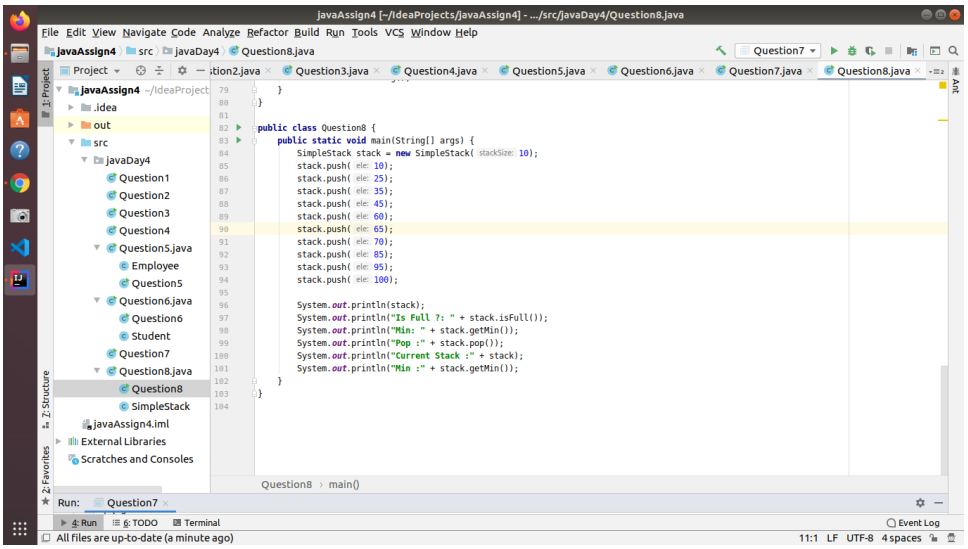
Question: 8:

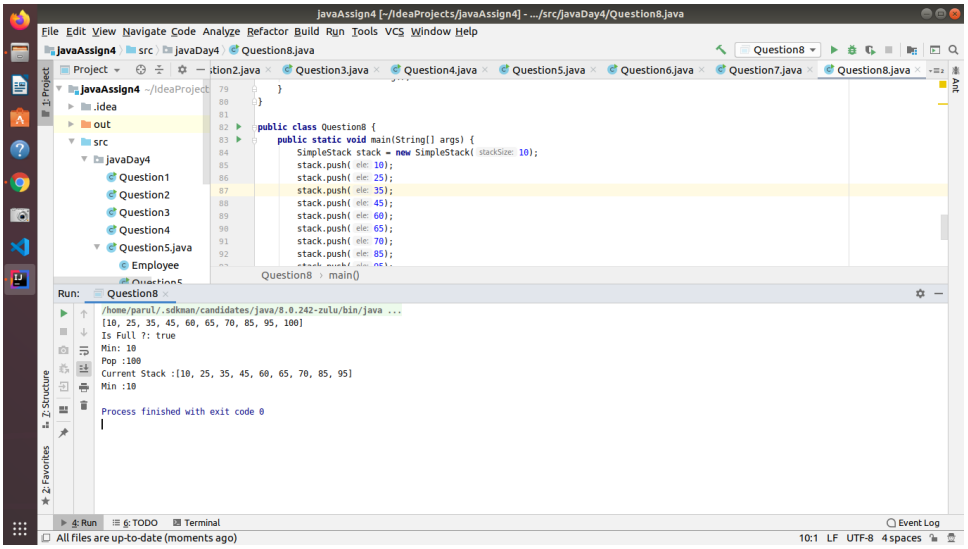
Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. (Expected complexity O(1))


javaAssign4 [~/IdeaProjects/javaAssign4] - .../src/javaDay4/Question8.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaAssign4 src javaDay4 Question8.java
Question7
Project
Question2.java Question3.java Question4.java Question5.java Question6.java Question7.java Question8.java
javaAssign4 ~/IdeaProject
.idea
out
src
javaDay4
Question1
Question2
Question3
Question4
Question5.java
Employee
Question5
Question6.java
Question6
Student
Question7
Question8.java
Question8
SimpleStack
javaAssign4.iml
External Libraries
Scratches and Consoles
package javaDay4;
import java.util.ArrayList;
import java.util.List;
class SimpleStack {
List<Integer> list = null;
private int stackSize;
private int min = Integer.MAX_VALUE;
private int top = -1;
SimpleStack(int stackSize) {
this.stackSize = stackSize;
try{
if (stackSize < 1) {
throw new Exception("Stack size cannot be less than 1");
}
} catch (Exception e) {
e.printStackTrace();
}
list = new ArrayList<>(stackSize);
}
void push(int ele){
try{
if (!isFull()) {
top++;
list.add(ele);
Question8 main()
Run: Question7
4: Run 6: TODO Terminal Event Log
All files are up-to-date (a minute ago)
11:1 LF UTF-8 4 spaces


javaAssign4 [~/IdeaProjects/javaAssign4] - .../src/javaDay4/Question8.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaAssign4 src javaDay4 Question8.java
Question7
Project
Question2.java Question3.java Question4.java Question5.java Question6.java Question7.java Question8.java
javaAssign4 ~/IdeaProject
.idea
out
src
javaDay4
Question1
Question2
Question3
Question4
Question5.java
Employee
Question5
Question6.java
Question6
Student
Question7
Question8.java
Question8
SimpleStack
javaAssign4.iml
External Libraries
Scratches and Consoles
void push(int ele){
try{
if (!isFull()) {
top++;
list.add(ele);
min = Math.min(ele, min);
} else {
throw new Exception("Stack is full");
}
} catch (Exception e) {
e.printStackTrace();
}
}
boolean isFull() {
return top == stackSize-1;
}
boolean isEmpty(){
return top == -1;
}
int getMin(){
return min;
}
Integer pop(){
try{
if (!isEmpty()) {
int popped = list.remove(top);
Question8 main()
Run: Question7
4: Run 6: TODO Terminal Event Log
All files are up-to-date (a minute ago)
11:1 LF UTF-8 4 spaces


javaAssign4 [~/IdeaProjects/javaAssign4] - .../src/javaDay4/Question8.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaAssign4 src javaDay4 Question8.java
Question7
Project
Question2.java Question3.java Question4.java Question5.java Question6.java Question7.java Question8.java
javaAssign4 ~/IdeaProject
.idea
out
src
javaDay4
Question1
Question2
Question3
Question4
Question5.java
Employee
Question5
Question6.java
Question6
Student
Question7
Question8.java
Question8
SimpleStack
javaAssign4.iml
External Libraries
Scratches and Consoles
Integer pop(){
try{
if (!isEmpty()) {
int popped = list.remove(top);
if(popped==min){
changeMin();
}
top--;
return popped;
} else {
throw new Exception("Stack is empty");
}
} catch (Exception e) {
e.printStackTrace();
}
return null;
}
private void changeMin(){
min = Integer.MAX_VALUE;
for (Integer ele : list) {
min = Math.min(min,ele);
}
}
@Override
public String toString() {
return list.toString();
}
}
Question8 main()
Run: Question7
4: Run 6: TODO Terminal Event Log
All files are up-to-date (a minute ago)
11:1 LF UTF-8 4 spaces


javaAssign4 [~/IdeaProjects/javaAssign4] - .../src/javaDay4/Question8.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaAssign4 src javaDay4 Question8.java
Question7
Project
Question2.java Question3.java Question4.java Question5.java Question6.java Question7.java Question8.java
javaAssign4 ~/IdeaProject
.idea
out
src
javaDay4
Question1
Question2
Question3
Question4
Question5.java
Employee
Question5
Question6.java
Question6
Student
Question7
Question8.java
Question8
SimpleStack
javaAssign4.iml
External Libraries
Scratches and Consoles
79 }
80 }
81
82 public class Question8 {
83 public static void main(String[] args) {
84 SimpleStack stack = new SimpleStack(stackSize: 10);
85 stack.push(ele: 10);
86 stack.push(ele: 25);
87 stack.push(ele: 35);
88 stack.push(ele: 45);
89 stack.push(ele: 60);
90 stack.push(ele: 65);
91 stack.push(ele: 70);
92 stack.push(ele: 85);
93 stack.push(ele: 95);
94 stack.push(ele: 100);
95
96 System.out.println(stack);
97 System.out.println("Is Full ?: " + stack.isFull());
98 System.out.println("Min: " + stack.getMin());
99 System.out.println("Pop :" + stack.pop());
100 System.out.println("Current Stack :" + stack);
101 System.out.println("Min :" + stack.getMin());
102 }
103 }
104
Question8 main()
Run: Question7
4: Run 6: TODO Terminal Event Log
All files are up-to-date (a minute ago)
11:1 LF UTF-8 4 spaces


javaAssign4 [~/IdeaProjects/javaAssign4] - .../src/javaDay4/Question8.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaAssign4 src javaDay4 Question8.java
Question8
Project
Question2.java Question3.java Question4.java Question5.java Question6.java Question7.java Question8.java
javaAssign4 ~/IdeaProject
.idea
out
src
javaDay4
Question1
Question2
Question3
Question4
Question5.java
Employee
79 }
80 }
81
82 public class Question8 {
83 public static void main(String[] args) {
84 SimpleStack stack = new SimpleStack(stackSize: 10);
85 stack.push(ele: 10);
86 stack.push(ele: 25);
87 stack.push(ele: 35);
88 stack.push(ele: 45);
89 stack.push(ele: 60);
90 stack.push(ele: 65);
91 stack.push(ele: 70);
92 stack.push(ele: 85);
Question8 main()
Run: Question8
/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
[10, 25, 35, 45, 60, 65, 70, 85, 95, 100]
Is Full ?: true
Min: 10
Pop :100
Current Stack :[10, 25, 35, 45, 60, 65, 70, 85, 95]
Min :10
Process finished with exit code 0
4: Run 6: TODO Terminal Event Log
All files are up-to-date (moments ago)
10:1 LF UTF-8 4 spaces