LAB ASSIGNMENT-1.1

Page No
Date

PROBLEM STATEMENT : To write a java program to display "Hello world ! I am learning Java."

CLASS DIAGRAM :

Hello
+ main (String [] args): void

ALGORITHM :

1. Start
2. Create public class Hello.
3. Create the public method main inside Hello class with a void return type and String [] arguments.
4. Print out the required string using System.out.println method.
5. End the method and class.
6. Stop.

INPUT, OUTPUTS AND ERROR HANDLING :

Inputs for the program	N/A
Expected outputs from the program	"Hello world! I am learning Java."
Error Handling	N/A

Teacher's Signature .....

TEST CASES AND RESULTS :

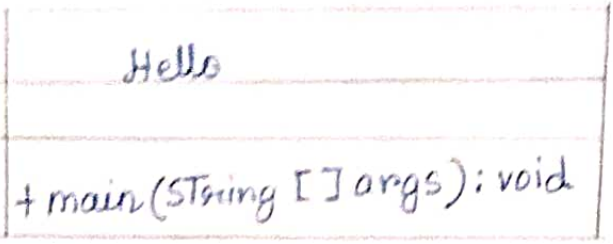
SL.	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	N/A	"Hello world ! I am learning Java."	"Hello world ! I am learning Java."	OK as expected and observed outputs are same.

CONCLUSIONE :

Through this programme we learnt how to declare classes, the main method of the Java program, and how to print out Strings in Java.

(Q2) PROBLEM STATEMENT : To write a Java program to input the user name and greet the user.

CLASS DIAGRAM :



ALGORITHM :

1. Start
2. Import the package java.util.Scanner to use the Scanner class.
3. Declare the public class Hello and inside it declare the main method with void return type and String[] arguments.
4. Create a new object of Scanner class named sc.
5. Using sc, input the user name and store it in a String variable named name.
6. Print out "Hello" and concatenate it with name.

Page No.
Date

EXPT. NO.

7. Use sc.close() method to stop the object sc.
8. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	name (String)
Expected outputs from the program	"Hello <name>!"
Error Handling	Input name should be a valid String

TEST CASES AND RESULTS :

SL.	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter name "Mr. James Gosling"	"Hello Mr. James Gosling!"	"Hello Mr. James Gosling!"	OK as expected and observed outputs are same.
2.	Enter name "Kabir Singh"	"Hello Kabir Singh!"	"Hello Kabir Singh!"	OK as expected and observed outputs are same.

CONCLUSION :

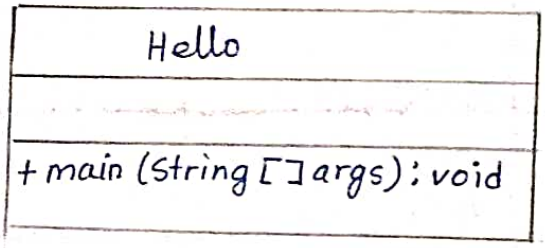
Through this assignment we understood how to use Scanner class in Java to take String inputs, and how to concatenate different strings into a single String for outputs.

Teacher's Signature .....

(Q3) PROBLEM STATEMENT :

To write a Java program to input marks of 2 different subjects and display their average, ignoring the fractional part.

CLASS DIAGRAM :



ALGORITHM :

1. Start
2. Import the package java.util.Scanner to use the Scanner class.
3. Declare public class Hello and inside declare the public main method with void return type and String [] parameters.
4. Create an object sc of the Scanner class.
5. Use sc.nextInt() method twice to input the marks and store them in the int variables n1 and n2.
6. Print out (n1 + n2)/2, i.e. their average.
7. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	n1 (int), n2 (int)
Expected outputs from the program	average (int)
Error Handling	Display error when n1<0 or n2<0

Page No
Date

EXPT NO.

TEST CASES AND RESULTS :

SL.	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter n1 as 50 and n2 as 31	40	40	OK as expected and observed outputs are same.
2.	Enter n1 as 10 and n2 as 20	15	15	OK as expected and observed outputs are same.
3.	Enter n1 as 21 and n2 as 12	16	16	OK as expected and observed outputs are same.
4	Enter n1 as 55 and n2 as 45	50	50	OK as expected and observed outputs are same.

CONCLUSION :

Through this assignment we understood how to use Scanner class in Java to take integer inputs, and how to use mathematical operators to compute average of numbers.

(Q4) PROBLEM STATEMENT :

To write a Java program to create a Rectangle class and a method to compute its area.

P.T.O

Teacher's Signature .....

CLASS DIAGRAM :

Rectangle
+ length : double + breadth : double
+ Rectangle (double) : void + Compute Area ( ) : double

ALGORITHM :

1. Start
2. Create a public class Rectangle having public double variable length and breadth.
3. Create a constructor method Rectangle ( ) with double parameters to assign the length and breadth values.
4. Create a method named compute Area ( ) which computes the area of the rectangle by multiplying the length and breadth and returns a double value.
5. Stop.

INPUTS , OUTPUTS AND ERROR HANDLING :

Inputs for the program	l (double) , b (double)
Expected outputs from the program	area (double)
Error Handling	Display error when l<=0 or b<=0

Page No.
Date

EXPT. NO.

TEST CASES AND RESULTS :

Sl	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter l as 2.0 and b as 3.0	6.0	6.0	OK as expected and observed outputs are same
2.	Enter l as 21.0 and b as 12.5	262.5	262.5	OK as expected and observed outputs are same.

CONCLUSION :

Through this assignment we understood how to use constructor methods to assign values to class variables, and how to create methods inside classes which involve computation with the class variables.

SUPER

Teacher's Signature .....

LAB ASSIGNMENT - 2.1

(9) PROBLEM STATEMENT :

To write a Java program to input a number (n) and display the natural numbers from 1 to n along with the running totals.

CLASS DIAGRAM :

CalcRunningTotal
+main (String[] args): void

ALGORITHM :

1. Start
2. Import the package java.util.Scanner to use the Scanner class.
3. Create a public class CalcRunningTotal and declare the public method main() in it with String[] parameters.
4. Create an object sc of Scanner class, and use it to input an integer and store it in int variable n.
5. Check if n < 1. If yes, print "ERROR" and terminate code. Else, continue.
6. Create an int variable sum = 0 and i = 1.
7. We add i to sum and print out i and sum separated by spaces.
8. Increment i by 1 and continue step 6 while i <= n.
9. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	n (int)
Expected outputs from the program	Natural numbers from 1 to n and the running totals (int)
Error Handling	Display error when n < 1

Page No.
Date

EXPT. NO.

TEST CASES AND RESULTS :

SL.	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter n as 3	1 1 2 3 3 6	1 1 2 3 3 6	OK as expected and observed outputs are same.
2	Enter n as 0	"ERROR"	"ERROR"	OK as expected and observed outputs are same.

CONCLUSION :

Through this assignment we understood how to use control flow in our code (if .. else statements) as well as how to use while loops for iteration with condition.

(Q2)

PROBLEM STATEMENT :

To write a Java program to input a number and display the reverse of that number.

CLASS DIAGRAM :

Reverse
+main(String[] args): void

Shilpi SUPER

Teacher's Signature .....

ALGORITHM :

1. Start
2. Import the package java.util.Scanner to use the Scanner class.
3. Create a public class Reverse containing the public main() method with String [ ] arguments.
4. Create an object sc of the Scanner class to take an integer input and assign it to the int variable num.
5. Initialise int rev = 0
6. Put rev as (num % 10) and print the digit. Then put num = num / 10
7. Repeat the above step while num > 0.
8. Stop

INPUTS , OUTPUTS AND ERROR HANDLING :

Inputs for the program	num (int)
Expected outputs from the program	Reverse of num (int)
Error Handling	Display error when num is not a number.

TEST CASES AND RESULTS :

Sl.	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter num as 173	371	371	OK as expected and observed outputs are same.
2.	Enter num as 256	652	652	OK as expected and observed outputs are same.

Page No.
Date

EXPT. NO.

CONCLUSION :

Through this assignment we understood how to use loops in our codes (while loop) using a loop variable and a particular terminating condition.

Shilpi SUPER

Teacher's Signature .....

LAB ASSIGNMENT – 2.2

(91) PROBLEM STATEMENT :

To write a java program to create a Rectangle class and a method to display its area.

CLASS DIAGRAM :

Rectangle
+ length : double + breadth : double
+ Rectangle (double l, double b) + compute Area ( ) : double

ALGORITHM :

1. Start
2. Create public class Rectangle
3. Create data items length and breadth of type double.
4. Create the public constructor method Rectangle inside the class to assign the values of the length and breadth.
5. Create the public method compute Area ( ) to return the area of the rectangle (length * breadth).
6. End the class.
7. Stop

INPUTS , OUTPUTS AND ERROR HANDLING :

Inputs for the program	length (double) , breadth (double)
Expected outputs from the program	Area of the rectangle (double)
Error handling	The length and breadth inputs should be positive numbers (>0)

Page No.
Date

EXPT NO.

TEST CASES AND RESULTS :

Sl.	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1	Enter l as 2.0 and b as 3.0	6.0	6.0	OK as expected and observed outputs are same.
2.	Enter l as 21 and b as 12.5	262.5	262.5	OK as expected and observed outputs are same.

CONCLUSIONS :

Through this program we learnt how to create parameterized constructor for a class and to create public methods inside the class which return some value based on the data items in the class.

(Q2) ## PROBLEM STATEMENT :

To write a java method fact (int n) to return factorial value for small n within 20

CLASS DIAGRAM :

Test
+ main(String [] args) : void

SUPER

Teacher's Signature .....

ALGORITHM :

1. Start
2. Create public method fact having parameter as int n and return type as long.
3. If n is negative, return -1, Else, continue. (Error code)
4. If n=0, return 1. Else, continue.
5. Recursively return n * fact (n-1).
6. End the method.
7. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	n (int)
Expected outputs from the program	factorial of n (long)
Error Handling	Display -1 when n<0

TEST CASES AND RESULTS :

Sl	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1	Enter n as 5	120	120	OK as expected and observed outputs are same
2	Enter n as 1	1	1	OK as expected and observed outputs are same
3	Enter n as 0	1	1	OK as expected and observed outputs are same
4	Enter n as 15	1307674368000	1307674368000	OK as expected and observed outputs are same.
5	Enter n as -5	-1	-1	OK as expected and observed outputs are same.

Page No.
Date

EXPT. NO

CONCLUSION :

Through this program we learnt how to create parameterized constructor for a class and to create public methods inside the class which return some value based on the data items in the class.

(Q3)

PROBLEM STATEMENT :

To write a java program to input subject code and subject name and display them.

CLASS DIAGRAM :

Course
+ String code + String name
+ Course ( )
+ Course (String c, String n)
+ get Course ( ) : String
+ set Course (String c, String n) : void

ALGORITHM :

1. Start
2. Create public class Course having data members code and name of String type.
3. Create a non parameterized constructor Course () which initializes code and name to empty string "".

SUPER

Teacher's Signature .....

4. Create a parameterized constructor Course ( ) to assign inputted strings to code and name.
5. Create method get Course ( ) to return a string having subject code and name.
6. Create another method set Course ( ) to assign given values to code and name.
7. Stop

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	code (String), name (String)
Expected outputs from the program	"<code>	<name>"
Error Handling	Inputs should be valid strings

TEST CASES AND RESULTS :

Sl	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter: ES-CS201 Programming for Problem Solving PCC-CS503 Object Oriented Programming	Courses: ES-CS201 | Programming for Problem Solving PCC-CS503 | Object Oriented Programming	Courses: ES-CS201 | Programming for Problem Solving PCC-CS503 | Object Oriented Programming	OK as expected and observed outputs are same.

CONCLUSION :

Through this program we learnt how to create parameterized as well as non-parameterized constructor for a class and override them as necessary, as well as using accessor and mutator methods.

LAB ASSIGNMENT - 2.3

Page No. /
Date /

EXPT NO. 81

PROBLEM STATEMENT :

To write a java program on method overloading, by using a method to find square of an integer or decimal.

CLASS DIAGRAM :

Method Overload
+ square (int n) : int
+ square (double b) : double

ALGORITHM :

1. Start
2. Create public class Method Overload.
3. Create public method square() which takes integer argument n and return the square as integer (n * n).
4. Create another public method square() having same name but takes double argument b and returns the square as double (b * b).
5. End the class.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	n (int) or b (double)
Expected outputs from the program	square (int) or square (double)
Error Handling	Input should only be of integer or double type.

Shilpi SUPER

Teacher's Signature .....

TEST CASES AND RESULTS :

Sl	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter n as -2	Square of -2 : 4	Square of -2 : 4	OK as expected observed output are same.
2.	Enter b as 1.5	Square of 1.500000 : 2.25	Square of 1.500000 : 2.25	OK as expected observed output are same.

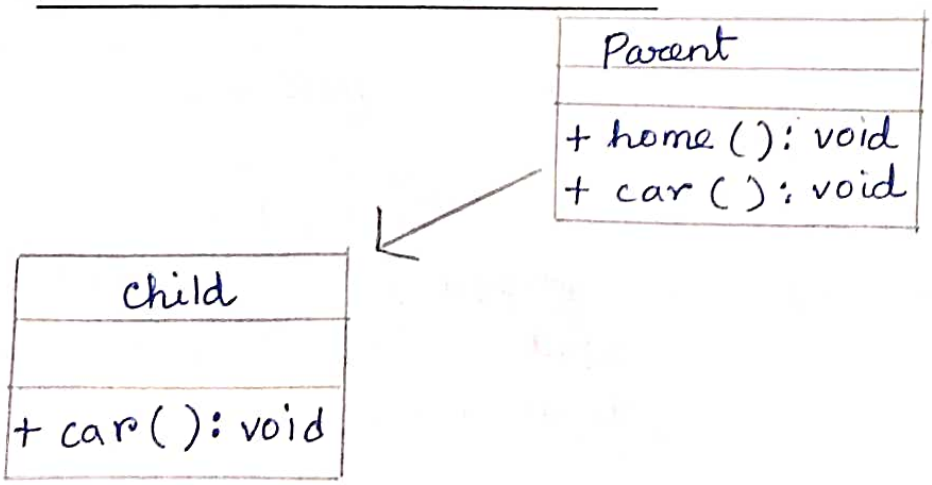
CONCLUSION :

Through this program we learnt how to use the Technique of method overloading on different methods with same method names to work on different types of inputs and produce desired outputs.

(Q2) ## PROBLEM STATEMENT :

To write a java program on method overriding, by using methods with same method names in both parent and child classes to print different strings.

CLASS DIAGRAM :



Page No. / Date /

EXPT. NO.

ALGORITHM :

1. Start
2. Create subclass Child which extends class Parent.
3. Create public method car() inside Child with void return type and it prints "New Car".
4. End the class.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	N/A
Expected outputs from the program	"Existing home New Car"
Error Handling	N/A

TEST CASES AND RESULTS :

Sl	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	N/A	Existing home New Car	Existing home New Car	OK as expected and observed outputs are same.

CONCLUSION :

Through this program we learnt how to use the technique of method overriding between parent and sub classes using the same method name, to display different outputs based on the objects of parent or sub class.

Shilpi SUPER

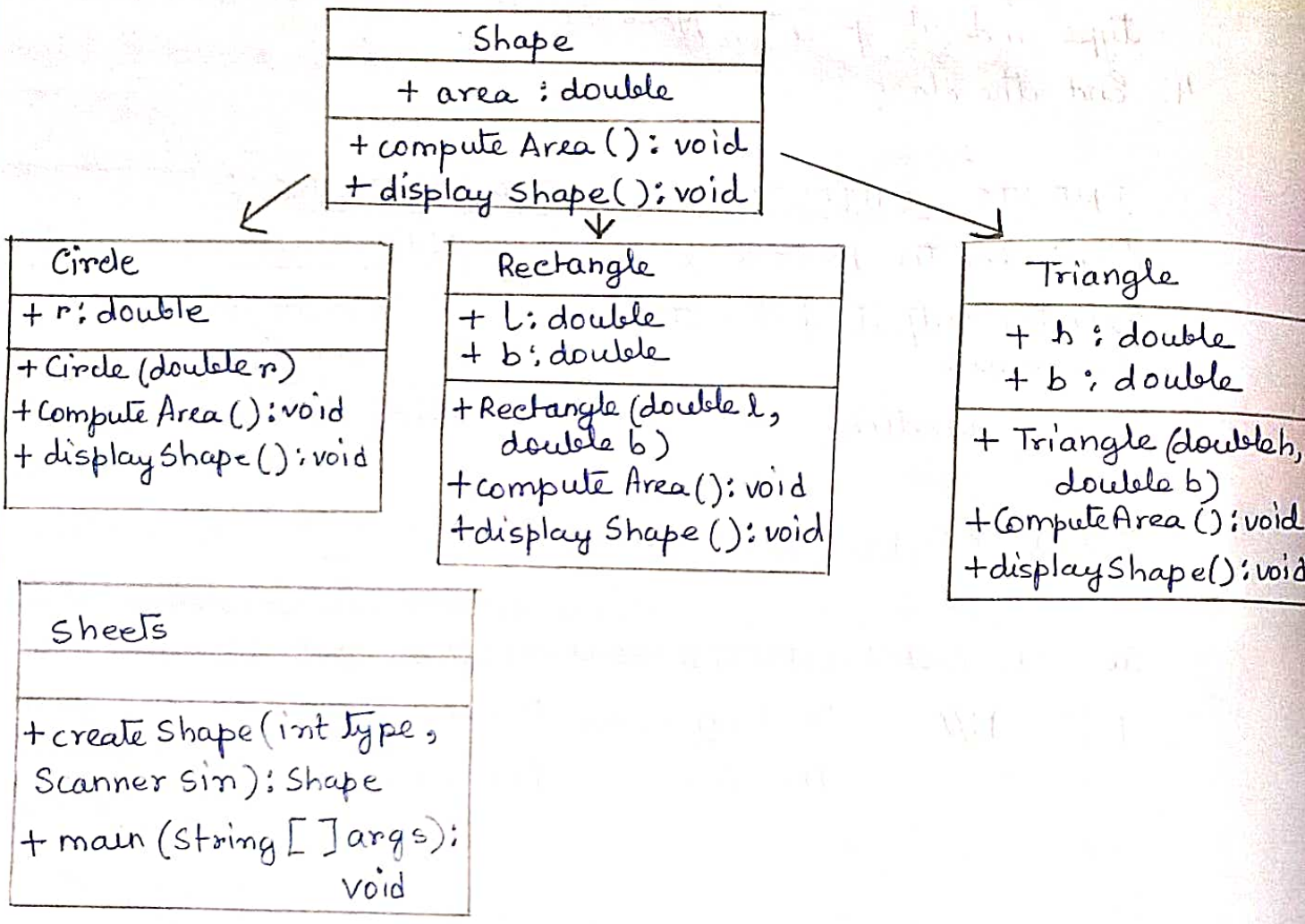
Teacher's Signature .....

(Q3)

PROBLEM STATEMENT :

To write a java program on method overriding, by using same method names in parent and sub classes to define area and structure of different shapes.

CLASS DIAGRAM :



ALGORITHM :

1. Start
2. Create public static method create Shape having integer type and Scanner sin arguments and a Shape return type.
3. Declare object obj of class Shape and two double variables a and b.

very descriptive Algo.

Page No
Date

EXPT. NO.

4. Create a switch statement with switch variable as type.
5. When case = 1, print out "Circle detected", Declare obj as an object of Circle subclass using parameterized constructor taking in next double input using sin. Also, use compute Area() method to calculate and display its area.
6. When case = 2, print out "Rectangle detected". Declare obj as an object of Rectangle subclass using parameterized constructor taking in next two double inputs using sin. Also, use compute Area() method to calculate and display its area.
7. When case = 3, print out "Triangle detected". Declare obj as an object of Triangle subclass using parameterized constructor taking in next two double inputs using sin. Also, use compute Area() method to calculate and display its area.
8. For default case, print out "Ignoring wrong type: <type>" and set obj to null.
9. Return obj.
10. End method.

Algo should be stepwise & short.

INPUTS, OUTPUTS AND ERROR HANDLING:

Inputs for the program	type (int) or double values.
Expected outputs from the program	Shape detection, displaying shape and calculating its area.
Error Handling	Inputs should only be non negative integers or double values (n > 0)

Shilpi SUPER

Teacher's Signature .....

TEST CASES AND RESULTS :

Sl.	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	2 5 6 9 1 5	Rectangle detected Ignoring wrong type : 9 Circle detected 0: Rectangle (5.00, 6.00) Area : 30.00 1: Circle (5.00) Area: 78.54	Rectangle detected Ignoring wrong type : 9 Circle detected 0: Rectangle (5.00, 6.00) Area : 30.00 1 : Circle (5.00) Area : 78.54	OK as expected and observed outputs are same.

CONCLUSION :

Through this program we learnt how to use switch statement for control flow in programs as well as the technique of method overriding between parent and sub classes using the same method name, to display different outputs based on the objects of parent or sub class.

LAB ASSIGNMENT – 3.1

Page No.
Date

EXPT NO 9.1

PROBLEM STATEMENT – To write a Java program to input numbers, sort them, and display maintaining 2 decimal places.

CLASS DIAGRAM :

Test
+ main (string[] args: void

ALGORITHM :

1. Start.
2. Create public class Test having main method.
3. Create an object sc of Scanner class.
4. While next input exists, keep inputting numbers with using sc and store them in an array named arr.
5. After each number is inputted at position i, while i > 0, check if arr[i] < arr[i - 1].
6. If yes, swap the elements, if no, pass.
7. In this way, the entire array is sorted each time a new element is added.
8. Run a for loop to print out the elements of the puppet, in %.2f format.
9. Stop.

SUPER

Teacher's Signature .....

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	Numbers (double)
Expected outputs from the program	Sorted array (double) in %.2f format
Error Handling	Inputs should be double values (no non-numbers)

TEST CASES AND RESULTS :

Sl	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1	21 10 5.7	5.70 10.00 21.00	5.70 10.00 21.00	OK as expected and observed outputs are same
2	15 7.5 30	7.50 15.00 30.00	7.50 15.00 30.00	OK as expected and observed outputs are same.

CONCLUSIONS :

Through this program we learnt how to take in input streams of numbers, how to declare arrays and input data to arrays, and how to do bubble sort in arrays.

(92)

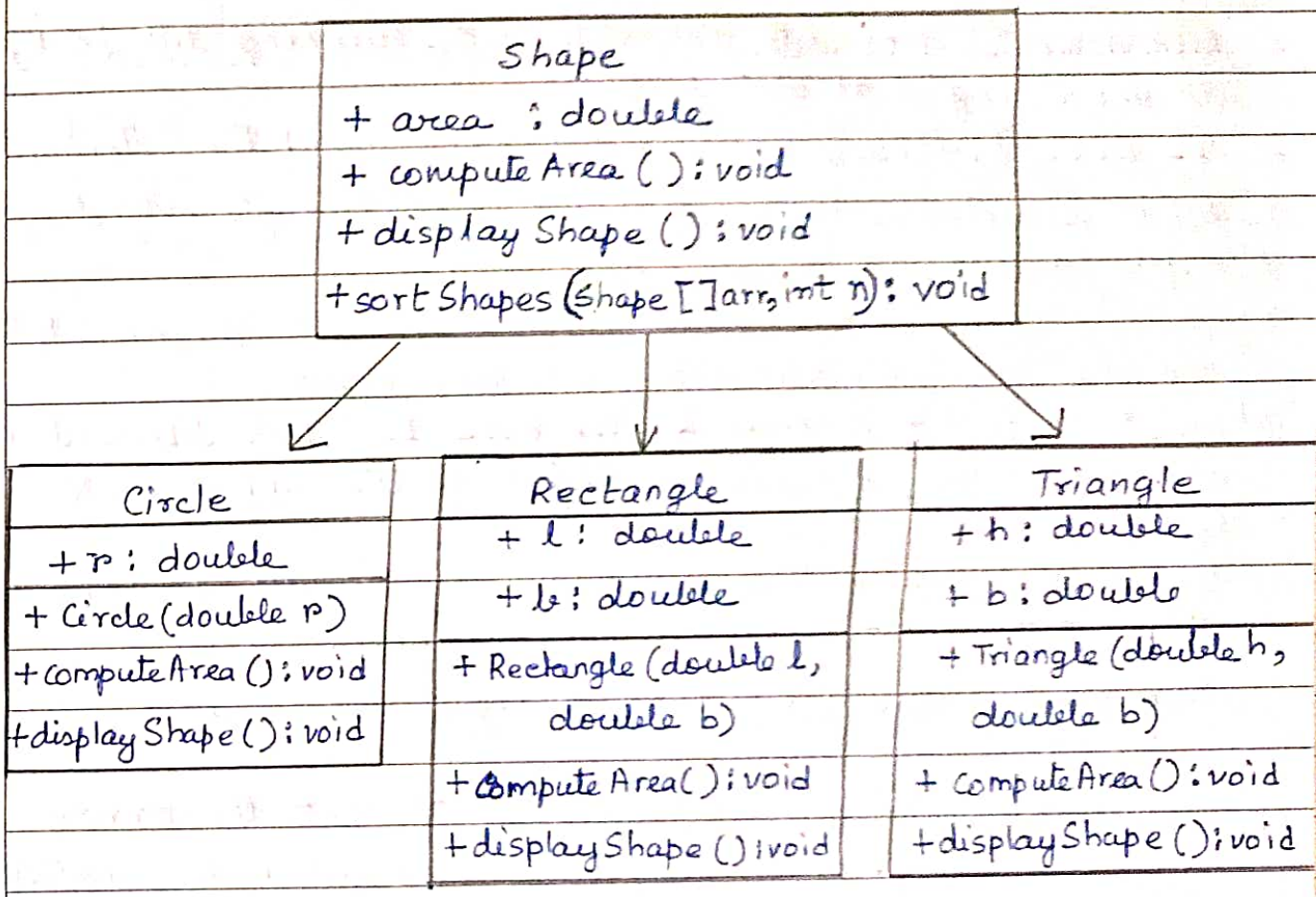
PROBLEM STATEMENT :

To write a Java program to deal with Shape entries, compute the Shape areas, sort entries and display their values in ascending order of areas.

Page No.
Date

EXPT. NO.

CLASS DIAGRAM :



Sheets
+ createShape (int type, Scanner sin) : Shape
+ main (String args [ ]) : void

ALGORITHM :

1. Start
2. Create a public class Shape with member and method computeArea (), displayShape ().

Teacher's Signature .....

3. Create static method sortShapes() which takes Shape[] arr and int n (length of arr) as arguments.
4. Run a for loop with loop variable $i = n-1$, running till $i > 0$, decrementing i at each loop.
5. Run a nested loop with variable $j = 0$, running till $j < i$, incrementing j at each loop.
6. If arr[j].area > arr[j+1].area, swap the elements.
7. Define subclasses Circle, Rectangle and Triangle extending Shape.
8. Define class Sheets having main method.
9. Inside, define static method createShape with arguments type(int) and sin(Scanner) and returns Shape object.
10. createShape() contains switch case to input different numbers and accordingly declare an object of the appropriate subclass of Shape.
11. Inside the main method, declare an array of Shape objects Shapes[] and an object sc of Scanner class.
12. Use createShape() method to input and create objects and store them in Shapes[].
13. Use static method sortShapes() to sort the array.
14. Run a for loop to display each element of the sorted array Shapes[].

INPUTS, OUTPUTS AND ERROR HANDLING:

Inputs for the program	type (int) or double values
Expected outputs from the program	Inserted shapes in ascending order of areas
Error Handling	Inputs should only be non-negative integers or double values ($n > 0$)

Page No.
Date

EXPT. NO.

TEST CASES AND RESULTS :

Sl	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	2 10	0: Circle (4.00)	0: Circle (4.00) Area:	OK as expected and
	8	Area: 50.27	50.27	observed
	1 4	1: Rectangle (10.00,	1: Rectangle (10.00,	outputs are same
		8.0) Area: 80.00	8.0) Area: 80.00	
2.	1 4	0: Rectangle (6.00,	0: Rectangle (6.00,	OK as expected
	2 6 4	4.00) Area: 24.00	4.00) Area: 24.00	and observed
	1 7	1: Triangle (10.00,	1: Triangle (10.00,	outputs are
	3 10	8.00) Area: 40.00	8.00) Area: 40.00	same.
	8	2: Circle (4.00)	2: Circle (4.00)	
		Area: 50.27	Area: 50.27	
		3: Circle (7.00)	3: Circle (7.00)	
		Area: 153.94	Area: 153.94	

CONCLUSION :

Through this assignment we understood how to use switch cases in Java, as well as how to sort an array of objects of a common class (different sub classes).

Teacher's Signature .....

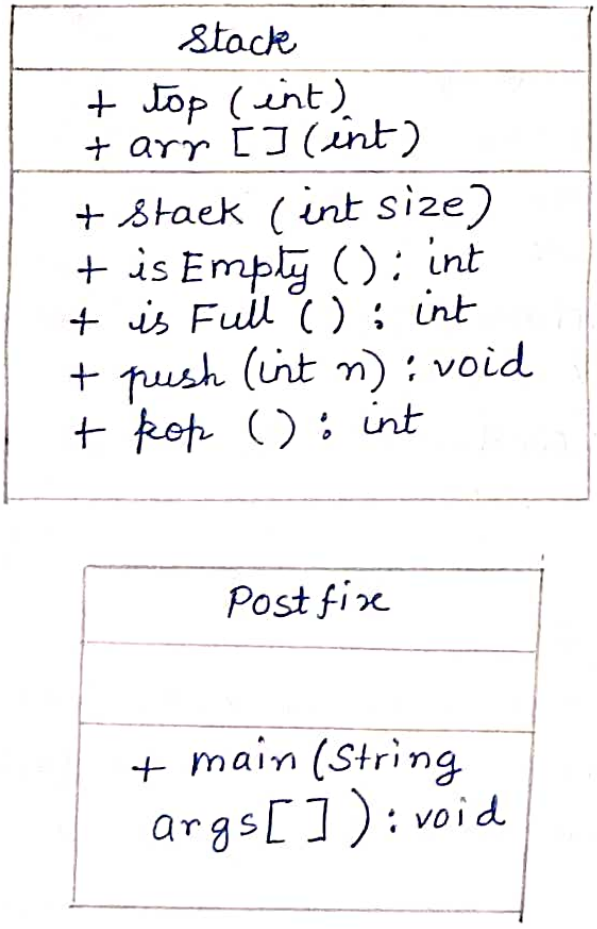
LAB ASSIGNMENT – 3.2

(91)

PROBLEM STATEMENT :

To write a java program for postfix expression evaluation based on stack - based solution with array implementation.

CLASS DIAGRAM :



ALGORITHM :

1. Start.
2. Declare a class Stack with int variables arr [] and top.
3. Declare a constructor to set the arr [] size and initialize top to -1.

Page No.
Date

EXPT. NO.

4. Create methods isEmpty(), isFull() to check whether stack is empty or full respectively.
5. Create methods push() to push an element into stack. Create method pop() to remove the top element of stack.
6. Declare main class Postfix containing main() method.
7. Use Scanner object sc to take in input line and split the components into a String array explist[].
8. Initialise Stack object st.
9. For each item in explist[], run a switch case to perform appropriate actions for the inputs in the stack.
10. Pop the last element of stack and print it as output.
11. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING:

Inputs for the program	Sequence of numbers and expressions (String)
Expected outputs from the program	ans (int)
Error Handling	Input string should not contain alphabets or special symbols.

Shilpi SUPER

Teacher's Signature .....

TEST CASES AND RESULTS :

Sl	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter input as 2 3 +	5	5	OK as expected and observed outputs are same.
2	Enter input as 1 -3 +	2	2	OK as expected and observed outputs are same.
3	Enter input as 5 2 3 * +	11	11	OK as expected and observed outputs are same

CONCLUSION :

Through this program we learnt how to use stack data structure using arrays and using switch cases to perform multiple choice operations.

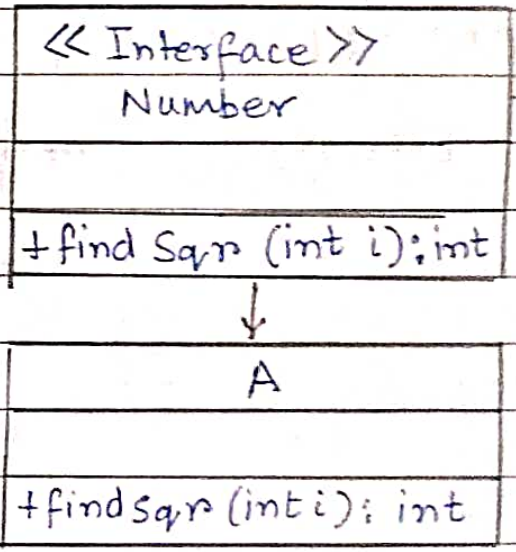
LAB ASSIGNMENT— 4.1

Page No.
Date

EXPT. NO. 81

PROBLEM STATEMENT : To write a java programme to complete the class A declaration implementing Number interface.

CLASS DIAGRAM :



ALGORITHM :

1. Start
2. Declare a class A which implements Number Interface
3. Override the method findSqr () inside it to find square of a number.

Teacher's Signature .....

4. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	i (int)
Expected outputs from the program	square of i (int)
Error Handling	Input should be a valid integer.

TEST CASES AND RESULTS :

SL	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter input as 5	25	25	OK as expected and observed outputs are same.
2.	Enter input as 6	36	36	OK as expected and observed outputs are same
3.	Enter input as 13	169	169	OK as expected and observed outputs are same

CONCLUSIONS :

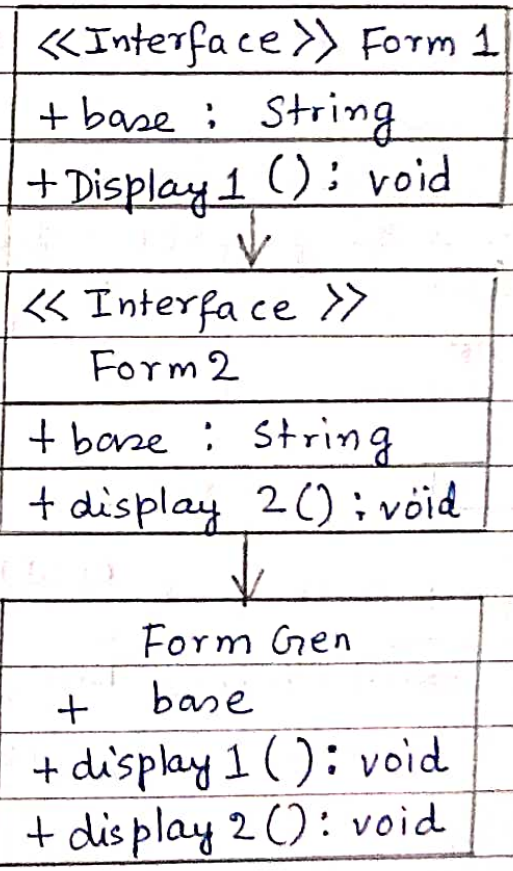
Through this program we learnt how to use interfaces which can be inherited in classes and how to override methods in sub classes.

Page No.
Date

EXPT NO 02

PROBLEM STATEMENT : To write a java program for dealing with different types of forms and ensure outputs as in the example.

CLASS DIAGRAM :



Demo Forms
+ main (String [ ] args) : void

Teacher's Signature .....

ALGORITHM :

1. Start.
2. Declare a method display 1() which prints out "What ? Form 1."
3. Declare a method display 2() which prints out "What ? Form 2."
4. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	N/A
Expected outputs from the program	Desired string
Error Handling	N/A.

TEST CASES AND RESULT

SL	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1	N/A	What? Form 1. What? Form 2.	What? Form 1 What? Form 2.0	OK as expected and observed outputs are same.

CONCLUSIONS :

Through this program we learnt how to use interfaces which can be inherited in classes and how to override methods in sub classes.

Page No.
Date

EXPT NO 03

PROBLEM STATEMENT: To write a java program to input n and add n number of integers. In case of Input Mismatch Exception, display "ERROR".

CLASS DIAGRAM :

Sum
+main(String[] args) : void

ALGORITHM :

1. Start.
2. Declare a class sum having main() method.
3. Open a try block.
4. Use Scanner object sc to input a number.
5. Run a loop from 0 to n, adding every new integer scanned to a sum variable.
6. Print out sum.
7. In case of any exception e, catch it and print out "ERROR".
8. Stop.

SUPER

Teacher's Signature .....

INPUTS , OUTPUTS AND ERROR HANDLING :

Inputs for the program	Number of inputs (int) and inpu numbers (int)
Expected outputs from the program	sum (int)
Error Handling	Inputs should be a valid integers.

TEST CASES AND RESULTS :

Sl.	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter input as 3 10 20 5	35	35	OK as expected and observed outputs are same
2	Enter input as 2 3 -3	0	0	OK as expected and observed outputs are same.
3.	Enter input as 3 -1 -5 -6	-12	-12	OK as expected and observed outputs are sam

CONCLUSIONS :

Through this program we learnt how to handle exceptions and errors using try-catch blocks and make custom error messages for the user.

LAB ASSIGNMENT- 5.1

Page No.
Date

EXPT NO (Q1)

PROBLEM STATEMENT : To write a java program for inputting a few numbers to populate an array list, sort it in descending order and perform a binary search on the array list.

CLASS DIAGRAM :

List Demo
+ populate List (Scanner sc, Array List <Integer>al): void
+ display List (string str, Array List <Integer>al): void
+ sort List Dese (Array List <Integer> al) : void
+ binsearch (Array List <Integer> al, int key) : void
+ main (string [ ] args) : void

ALGORITHM :

1. Start.
2. In class List Demo, create a static method populate List() which inputs a series of numbers and populates the array list.
3. Create a static method display List () to display the created array list along with a custom prefix.
4. Create a static method sort List Desc () to sort the list in descending order using reverse order sorting of Collections class.

SUPER

Teacher's Signature .....

5. Create a method binSearch() to search for a particular key in the list, using the divide and conquer approach and return its index.

6. If the index returned >= 0, then print the position. Else, print "Element not found.".

7. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	Series of numbers (int) and a key (int)
Expected outputs from the program	Print original List (String), sorted List (String) and the index of element (int)
Error Handling	Input numbers should be valid integers.

TEST CASES AND RESULTS :

Sl	TEST CASE	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter input as 45 23 67 12 23 ?	Original List : 45 23 67 12 Sorted List : 67 45 23 12 Position : 2	Original List : 45 23 67 12 Sorted List : 67 45 23 12 Position : 2	OK as expected and observed outputs are same.
2.	Enter input as 56 43 85 27 62	Original list: 56 43 85 27 Sorted List : 85 56 43 27 Element Not Found.	Original List : 56 43 85 27 Sorted List : 85 56 43 27 Element Not Found	OK as expected and observed outputs are same.

Page No.
Date

EXPT. NO.

CONCLUSIONS :

Through this program we learnt how to use array list and concepts of Generic programming, and how to apply searching and sorting techniques using array list in Java.

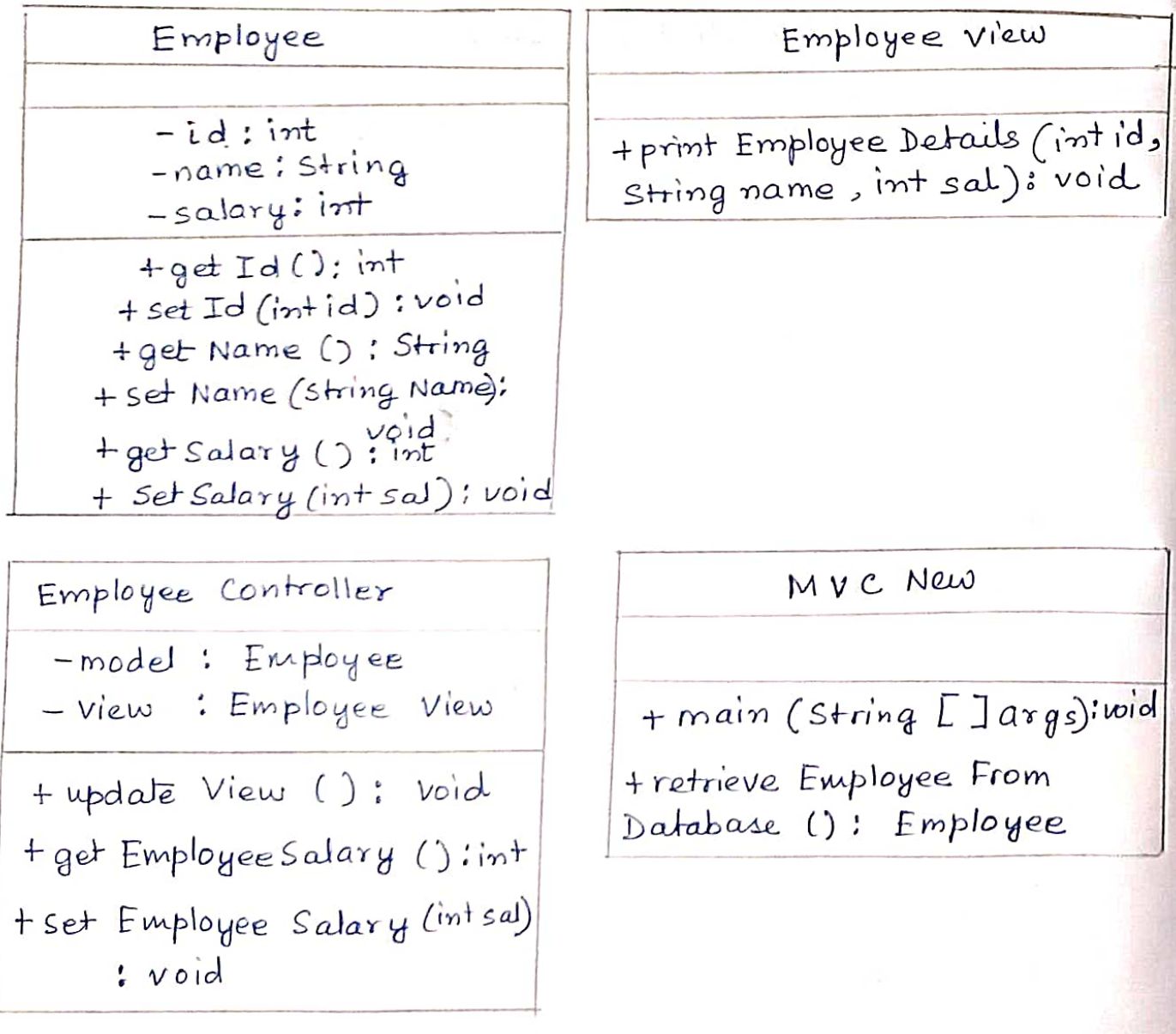
SUPER

Teacher's Signature .....

LAB ASSIGNMENT - 5.2

(Q1) PROBLEM STATEMENT : To write a program leveraging MVC architecture, that retrieves employee data, updates and displays view.

CLASS DIAGRAM :



Page No.
Date

EXPT NO

ALGORITHM :

1. Start.
2. Create a class Employee Controller with private data members model (Employee type) and view (Employee View type).
3. Create a public constructor of the class to initialize the values of model and view.
4. Create a public method updateView(), which calls the printEmployeeDetails() method of the view object with accessor methods of model, to display the details of the Student model.
5. Create a public method getEmployeeSalary(), which calls the method getSalary() of model object to return the private salary data of the model object.
6. Create a public method setEmployeeSalary() taking an argument sal, which calls the method setSalary(sal) of the model object to update its private data salary.
7. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	N/A
Expected outputs from the program	Employee details with old and updated salary.
Error Handling	N/A

SUPER

Teacher's Signature .....

TEST CASE AND RESULTS :

Sl	EXPECTED RESULT	OBSERVED RESULT	STATUS
	Id : 1012349 | Name : Ayushman Khurana | Salary : 150000 Id : 1012349 | Name : Ayushman Khurana | Salary : 165000	Id : 1012349 | Name : Ayushman Khurana | Salary : 150000 Id : 1012349 | Name : Ayushman Khurana | Salary : 165000	Ok as expected and observed results are same

CONCLUSIONS :

Through this program we learnt how to use MVC architecture to create programs in Java, and also how to use accessor and mutator methods to access and update private data members of classes respectively.

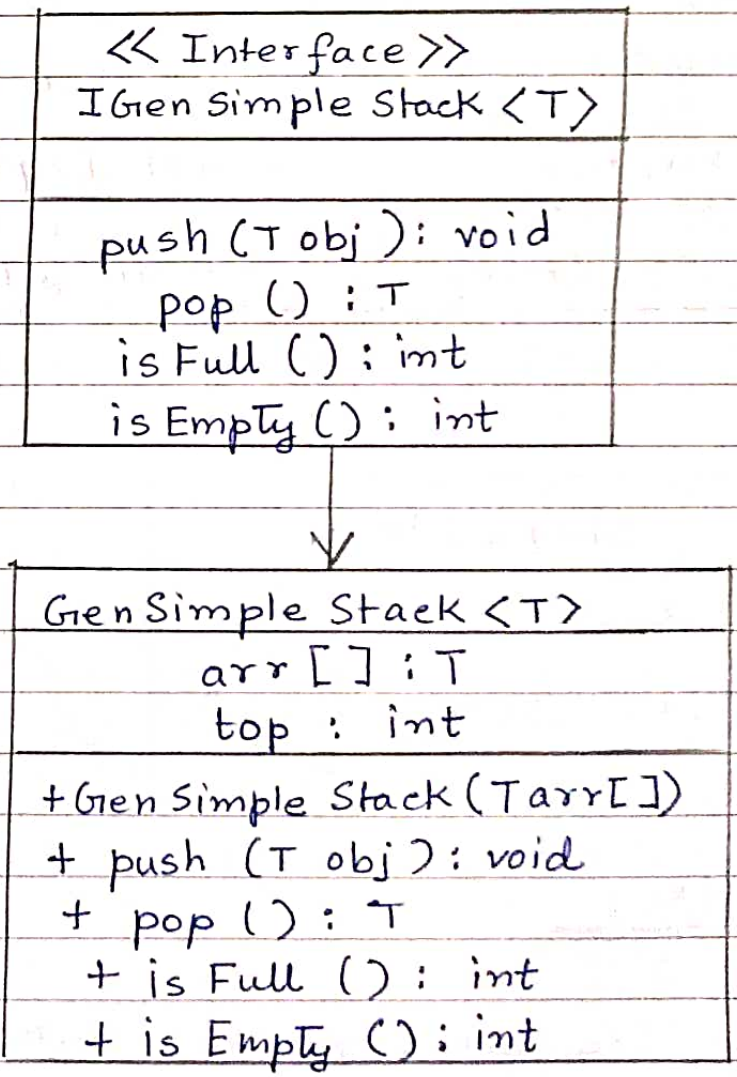
LAB ASSIGNMENT – 5.3

Page No.
Date

EXPT NO 8.1

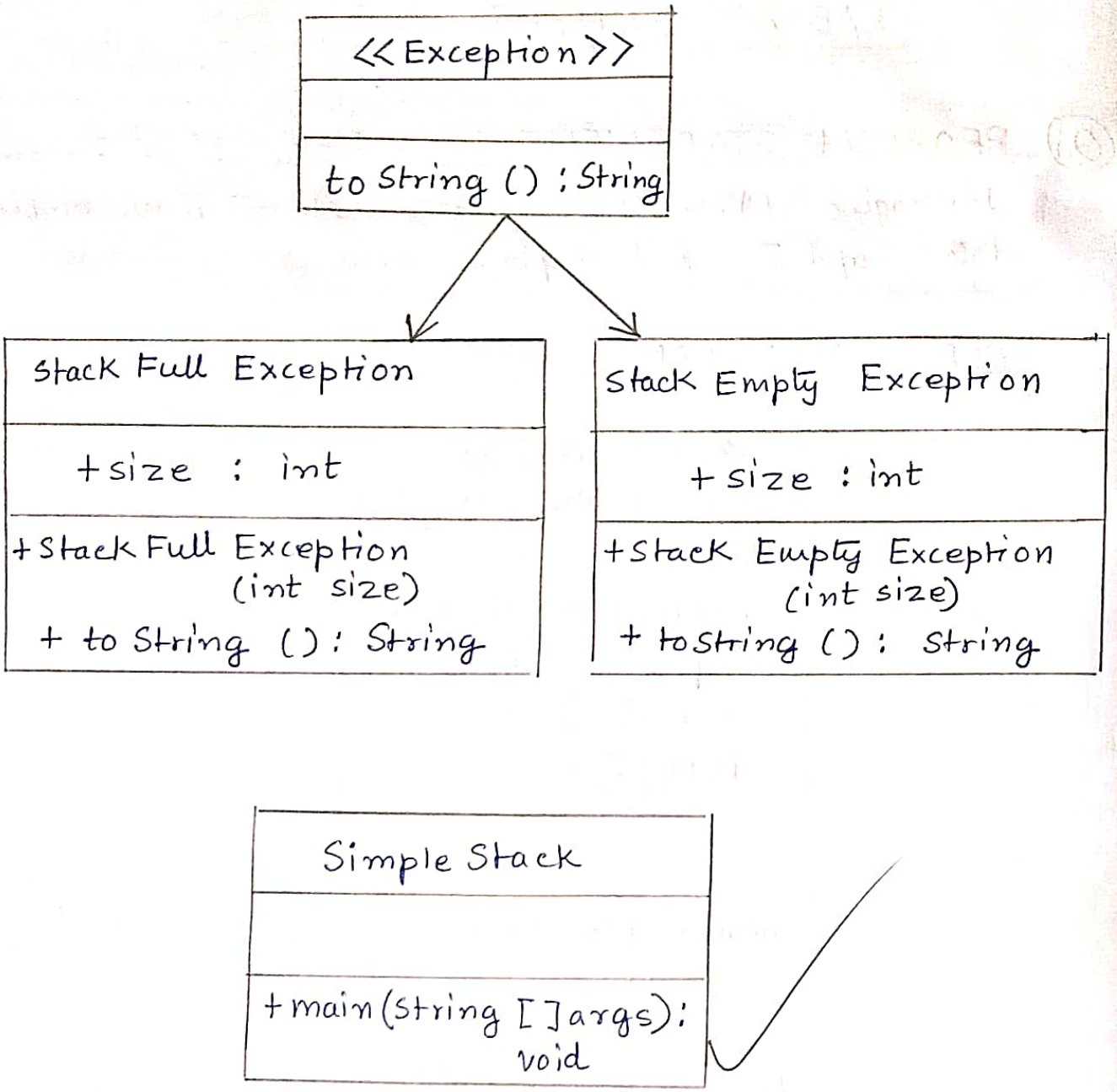
PROBLEM STATEMENT : To write a program leveraging MVC architecture, that retrieves employee data, updates and displays view.

CLASS DIAGRAM :



SUPER

Teacher's Signature .....



ALGORITHM :

1. Start.
2. Create an interface IGenSimpleStack with generic type T containing basic stack methods : push(), pop(), isFull(), and isEmpty().
3. Create a class GenSimpleStack with generic type T which

Page No.
Date

EXPT NO.

implements IGenSimpleStack. It creates an array based stack implementation.

4. Override the methods for push, pop, isEmpty and isFull, where push and pop throw exceptions.
5. Create classes StackEmptyException and StackFullException which implements Exception class and override the toString() method to display a custom error message.
6. Create the class SimpleStack which has the static main method.
7. Inside the main method, declare GenSimpleStack with String type and create a condition statement to push and pop elements based on number inputted by Scanner object.
8. Enclose the choices in a try-catch block to catch the possible exceptions thrown.
9. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING:

Inputs for the program	Choice (int) and elements (String)
Expected outputs from the program	Popped elements or exception message (both String)
Error Handling	Choices should be integer type.

SUPER

Teacher's Signature .....

TEST CASES AND RESULTS :

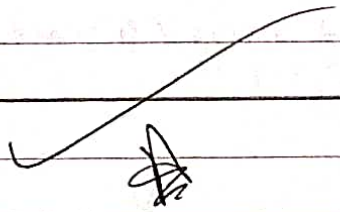
SL.	INPUT	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	2 0	[Empty]	OK as expected and observed results are same.	OK as expected and observed results are same.
2.	1 Tom 1 Jerry 2 2 0	Jerry Tom	Jerry Tom	OK as expected and observed results are same
3.	1 Tom 1 Jerry 1 Spike 1 Tyke 1 Toodle 1 Nibbles 2 2 0	[Full-5] Toodle Tyke	[Full-5] Toodle Tyke	OK as expected and observed results are same.

Page No.
Date

EXPT. NO.

CONCLUSIONS :

Through this program we learnt how to use generic programming in Java to implement data structures and how to throw user defined exceptions displaying custom messages in the output



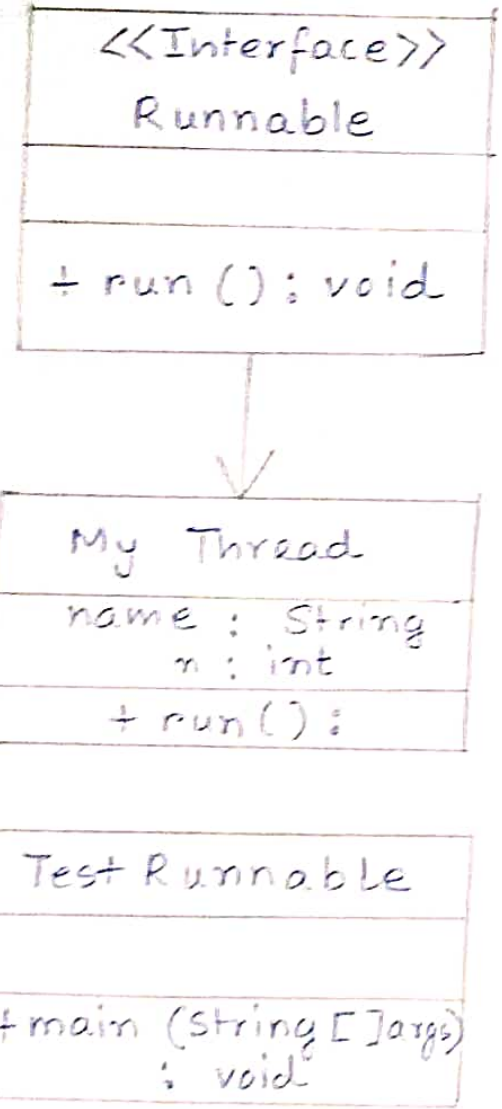
SUPER

Teacher's Signature .....

LAB ASSIGNMENT – 6.1

(Q1) **PROBLEM STATEMENT :** To write a program to compute the factorial of a number using Threading.

CLASS DIAGRAM :



ALGORITHM :

1. Start
2. Create a class My Thread which implements the Runnable interface containing name (String) and n (int).
3. Create a constructor for the class which assigns the values to name and n.

Page No.
Date

EXPT. NO.

4. Define the run method which to compute the value of the factorial of n and print the Thread name.
5. Print out the Thread ended message.
6. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING:

Inputs for the program	n (int)
Expected outputs from the program	Thread start and end message (string) and factorial (int)
Error Handling	Input should be integer.

TEST CASES AND RESULTS:

SL	INPUT	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	4	Main Thread : Started	Main Thread : Started	OK as
		Main Thread : Ended	Main Thread : Ended	expected
		Factorial Thread : started	Factorial Thread : Started	and observed
		Factorial Thread : 24	Factorial Thread : 24	results are
		Factorial Thread : Ended	Factorial Thread : Ended	same
2	-3	Main Thread : Started	Main Thread : Started	OK as
		Main Thread : Ended	Main Thread : Ended	expected
		Factorial Thread : Started	Factorial Thread : Started	and observed
		Factorial Thread : ERROR	Factorial Thread : ERROR	results are
		Factorial Thread : Ended	Factorial Thread : Ended	same.

SUPER

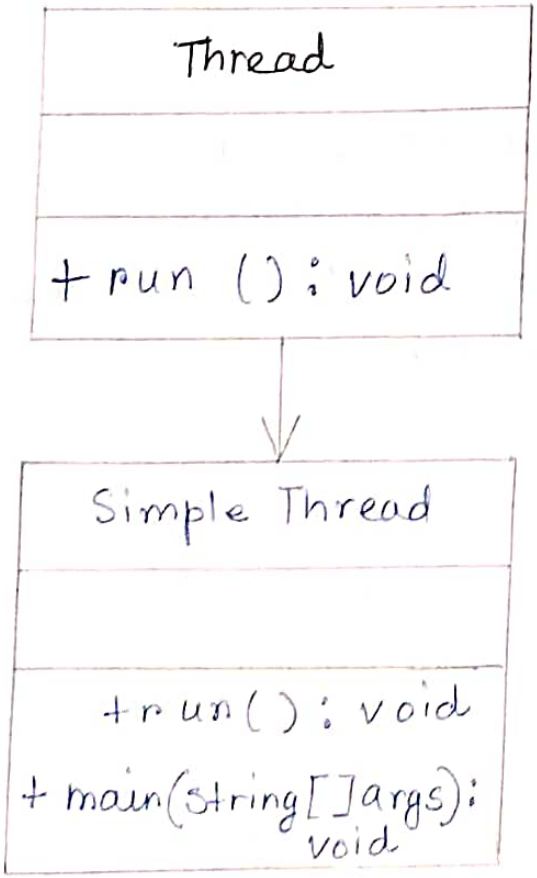
Teacher's Signature .....

CONCLUSIONS :

Through this program we learnt how to use Threading in Java by implementing Runnable interface, for simultaneous execution of processes, thereby reducing total execution time.

(Q2) **PROBLEM STATEMENT :** To write a program to run a thread using Thread superclass.

CLASS DIAGRAM :



Page No
Date

EXPT. NO.

ALGORITHM :

1. Start
2. Create a constructor to call the start () method within the Simple Thread class.
3. Create a method run () to print "Running Thread".
4. Stop.

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	N/A
Expected outputs from the program	"Running Thread"
Error Handling	N/A

TEST CASES AND RESULTS :

SL	INPUT	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	N/A	Running thread	Running thread	OK as expected and observed results are same

CONCLUSIONS : Through this program we learnt how to use threading in Java by extending Thread superclass, for simultaneous execution of processes, thereby reducing total execution time.

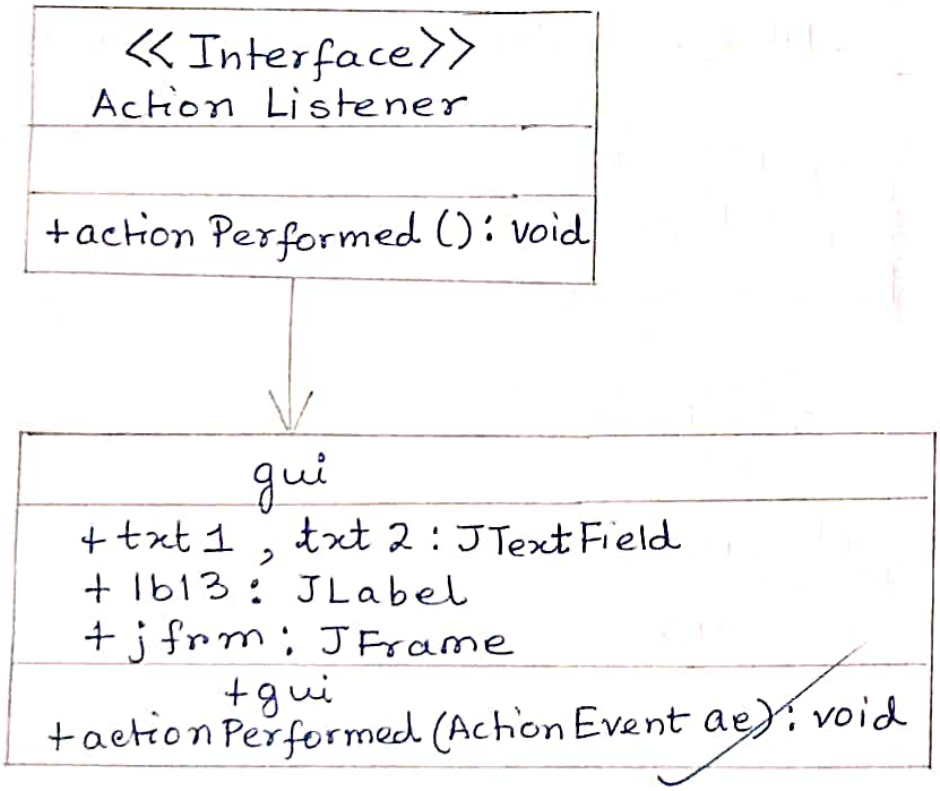
Teacher's Signature .....

SUPER

LAB ASSIGNMENT - 7.1

(Q1) **PROBLEM STATEMENT:** To write a program to develop a graphic window for capturing the admission counts of two streams and counts should be added and displayed. Validations checks for input values should be performed.

CLASS DIAGRAM:



ALGORITHM:

1. Start.
2. Import the GUI libraries in Java (java.awt, javax.swing).
3. Create a class gui which implements Action Listener.
4. Declare text fields txt1 and txt2, Label lbl3 and a frame jfrm for the GUI window.

Page No.
Date

EXPT NO

5. Create a constructor for the class, where we set the size of jfrm to 300 x 300.
6. Set the default close operation, set resizable to false.
7. Create new labels lbl1, lbl2 which read "CSE Count" and "CSBS Count" respectively.
8. Add text fields txt1 and txt2 along with the labels to input counts.
9. Create a submit button and add listener to this button.
10. Add all components to the jfrm and set its visibility to true.
11. Create actionPerformed() method, which checks validity of inputs in the text fields while pressing Submit.
12. If validation passed, the lock the fields and display Total. Else display appropriate Error Message.
13. Create a main method and create an object of gui class.
14. Stop.

P.T.O

Teacher's Signature .....

INPUTS, OUTPUTS AND ERROR HANDLING :

Inputs for the program	CSE count (int) and CSBS count (int).
Expected outputs from the program	Total count (int) or error message.
Error Handling	Inputs should be positive integers.

TEST CASES AND RESULTS :

Sl.	INPUT	EXPECTED RESULT	OBSERVED RESULT	STATUS
1.	Enter CSE count as 50 and CSBS Count as 40	Total : 90 (Text fields Locked)	Total : 90 (Text fields locked)	OK as expected and observed results are same.
2.	Enter CSE count as abc and CSBS count as 50	"Invalid input. Please enter integer value." (Text fields not locked)	"Invalid input. Please enter integer values." (Text fields not locked)	OK as expected and observed results are same.
3.	Enter CSE count as 50 and CSBS count as -5	"Intake cannot be negative. Please enter positive value." (Text fields not locked)	"Intake cannot be negative. Please enter positive value." (Text fields not locked)	OK as expected and observed results are same.

Page No.
Date

EXPT. NO.

CONCLUSIONS :

Through this program we learnt how to use Java packages to create GUI which are interactive We learnt how to perform operations with clicking buttons by implementing Action Listener.

SUPER

Teacher's Signature .....