**JAVA**

**Classes and Methods**

**Lab Exercise No:**14

**Exercise Objective(s):***Construction of an object*

**Exercise:***Create a class called Calculator which has 4 different methods add, diff, mul and div which*

*accepts two numbers as parameters. Create an object to access these methods and invoke*

*these methods with two numbers and display the result in the corresponding methods.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A calculator program which take two input numbers.

\*

\* Created Aniket Wattamwar

\*/

**public** **class** Calculator {

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

//Taking input two numbers

**int** x = Integer.*parseInt*(args[0]);

**int** y = Integer.*parseInt*(args[1]);

**int** ans = *add*(x,y);

System.***out***.println("The addition of the two numbers is "+ ans);

**int** sub = *subtract*(x,y);

System.***out***.println("The difference of the two numbers is "+ sub);

**int** divide = *divide*(x,y);

System.***out***.println("After dividing the two number. The result is "+ divide);

**int** mul = *multiply*(x,y);

System.***out***.println("The multiplication of the two numbers is "+ mul);

}

//addition of two numbers

**public** **static** **int** add(**int** x, **int** y) {

**return** x+y;

}

//Subtraction of two numbers

**public** **static** **int** subtract(**int** x, **int** y) {

**return** x-y;

}

//Division of two numbers

**public** **static** **int** divide(**int** x, **int** y) {

**return** x/y;

}

//Multiplication of two numbers

**public** **static** **int** multiply(**int** x, **int** y) {

**return** x\*y;

}

}

**Lab Exercise No:**15

**Exercise Objective(s):***Construction of an object, constructors*

**Exercise:***Create a class called Sample. Write a program to display the no of objects created for that*

*class or the no of times that class is instantiated.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):** *Use a static int variable and increment that variable inside the*

*constructor.*

**package** com.hsbc.solutions;

/\*

\* A program to display the no of objects created for that class or the no of times that class is instantiated.

\*

\* Created Aniket Wattamwar

\*/

**public** **class** CountObjects {

**static** **int** *i* =0;

CountObjects(){

*i*+=1; //increment the count when a new object is created

}

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

//creating objects

CountObjects obj1 = **new** CountObjects();

CountObjects obj2 = **new** CountObjects();

System.***out***.println("count is " + *i*);

}

}

**Lab Exercise No:**16

**Exercise Objective(s):***Construction of an object, this keyword, accessors(getters) and mutators(setters),*

*public and private access specifiers, instance and class member variables*

**Exercise:***Create a class called Student with the following details: RollNo, StudName, MarksInEng,*

*MarksInMaths and MarksInScience. Write getters and setters for the all variables. RollNo*

*should be automatically generatedwhenever a newstudent is added.*

*Create a class called Standard with 8 students’ details and write separate method for each of*

*the following tasks and invoke the same.*

1. *To display the entire roll no and the name of the students in the class in the ascending order of roll no.*
2. *To display the roll no and the name of the student who has got the highest percentage.*
3. *To display the roll no and the name of the student who scored highest mark*

*inmathematics.*

1. *To display the roll no and the name of the student in the ascending order of the total marks in mathematics and science alone.*
2. *To display the roll no, name, total marks, percentage and rank of all the students in the descending order of rank.*

**Recommended duration:***40Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**17

**Exercise Objective(s):***String class, String immutability*

**Exercise:***Write class that declares the following String.*

***“The quick brown fox jumps over the lazy dog”.***

*Perform the following modifications to the above string using appropriate methods.*

1. *Print the character at the 12th index.*
2. *Check whether the String contains the word “is”.*
3. *Add the string “and killed it” to the existing string.*
4. *Check whether the String ends with the word “dogs”.*
5. *Check whether the String is equal to “The quick brown Fox jumps over the lazy Dog”.*
6. *Check whether the String is equal to “*THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG*”.*
7. *Find the index position of the character “a”.*
8. *Find the last index position of the character “e”.*
9. *Find the length of the String.*
10. *Check whether the String matches to “The quick brown Fox jumps over the lazy Dog”.*
11. *Replace the word “The” with the word “A”.*
12. *Split the above string into two such that two animal names do not come together.*
13. *Print the animal names alone separately from the above string.*
14. *Print the above string in completely lower case.*
15. *Print the above string in completely upper case.*

**Recommended duration:***30Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A program about different String Manipulation

\*

\* Created Aniket Wattamwar

\*/

**public** **class** StringManipulations {

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

String str = "The quick brown fox jumps over the lazy dog";

System.***out***.println("Character at 12th index "+ str.charAt(11));

//checking if string contains 'is'

String word="";

**boolean** flag = **false**;

**for**(**int** i=0;i<str.length();i++)

{

**char** ch=str.charAt(i);

**if**(Character.*isLetter*(ch))

{

word=word+ch;

}

**else**

{

**if**(word.equals("is"))

{

System.***out***.println("Contains 'is'");

flag=**true**;

**break**;

}

**else**

{

word="";

}

}

}

**if** (flag==**false**)

System.***out***.println("Doesn not contain 'is'");

//adding 'and killed it' to existing string

String a=str+" and killed it";

System.***out***.println("After appending: "+a);

//checking if string ends with dogs

word="";

**for**(**int** i=str.length()-1;i<str.length();i--){

**char** ch=str.charAt(i);

**if**(Character.*isLetter*(ch)){

word=ch+word;

}

**else**{

**if**(word.equals("dog")){

System.***out***.println("Ends with 'dog'");

**break**;

}

**else**{

System.***out***.println("Does not end with 'dog'");

}

}

}

// Checking whether the String is equal to the given one

**if**(str.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

System.***out***.println("Equal");

**else**

System.***out***.println("Not Equal");

//Checking whether the String is equal to “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”

**if**(str.compareTo("THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG")==0)

System.***out***.println("Equal");

**else**

System.***out***.println("Not Equal");

//Finding the index position of the character 'a'

System.***out***.println(str.indexOf("a"));

//Finding the last index position of the character 'e'

System.***out***.println(str.lastIndexOf("e"));

//Finding the length of the String.

System.***out***.println(str.length());

//Checking whether the String matches to the given above

**if**(str.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

System.***out***.println("Equal");

**else**

System.***out***.println("Not Equal");

//Replacing the word “The” with the word “A”

word="";

String t="";

**for**(**int** i=0;i<str.length();i++)

{

**char** ch=str.charAt(i);

**if**(Character.*isLetter*(ch))

{

word=word+ch;

}

**else**

{

**if**(word.equals("The")||word.equals("the"))

{

t=t+" A";

word="";

}

**else**

{

t=t+" "+word;

word="";

}

}

}

System.***out***.println("After replacing "+ t);

//Splitting the above string into two such that two animal names do not come together

String d[]=str.split(" ",str.length());

System.***out***.println("After splitting");

**for**(**int** i=0;i<d.length;i++)

{

System.***out***.println(d[i]);

}

//Printing the animal names alone separately from the above string

t= str+ " ";

word="";

System.***out***.println("Animal Names");

**for**(**int** i=0;i<t.length();i++)

{

**char** ch=t.charAt(i);

**if**(Character.*isLetter*(ch))

{

word=word+ch;

}

**else**

{

**if**(word.equals("fox")||word.equals("dog"))

{

System.***out***.println(word);

word="";

}

**else**

{

word="";

}

}

}

//Printing in lowercase

System.***out***.println("String in lowercase: "+ str.toLowerCase());

//Printing in upper case

System.***out***.println("String in uppercase: "+ str.toUpperCase());

}

}

**Lab Exercise No:**18

**Exercise Objective(s):***String class, String immutability*

**Exercise:***Write a program to demonstrate the difference between equals and == operator with*

*appropriate example.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A program to show how the equals() and == operator are different

\*

\* Created Aniket Wattamwar

\*/

**public** **class** OperatorExample {

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

//create two strings using new keyword

String s1 = **new** String("abc");

String s2 = **new** String("abc");

//using the == and equals() to see what result it gives

System.***out***.println("== used and the value is " + s1==s2);

System.***out***.println("equals() used and the value is " +s1.equals(s2));

//equals() compares the value in s1 and s2.

//== compares the addresses of the two hence the output is false

}

}

**Lab Exercise No:**19

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to declare an array with 8 elements and copy the 8 elements into another*

*array and display the same.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A program to copy the elements from one array to another and displaying the same

\*

\* Created Aniket Wattamwar

\*/

**public** **class** CopyArrayElements {

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

//creating an array

**int** arr[] = {3,75,12,98,23,96,211,63};

//copy of the array with the same length

**int** arrCopy[] = **new** **int**[arr.length];

//looping and copying all the arrays in another array

**for**(**int** i=0;i<arr.length;i++) {

arrCopy[i]= arr[i];

System.***out***.print(arr[i]+ ",");

}

//printing the arrCopy elements

**for**(**int** i=0;i<arrCopy.length;i++) {

System.***out***.println("Array Elements are: " + arrCopy[i]);

}

}

}

**Lab Exercise No:**20

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to display the sum and the average of elements in the array.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A program to calculate the sum and avg of the array elements

\*

\* Created Aniket Wattamwar

\*/

**public** **class** SumAndAvg {

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

//creating an array

**int** arr[] = {3,75,12,98,23,96,211,63};

//variable to store the sum

**int** sum=0;

//calculate the sum of all the elements in the array

**for**(**int** i=0;i<arr.length;i++) {

sum=sum+ arr[i];

}

System.***out***.print(sum + "\n");

//calculate the average of the sum

System.***out***.println("The average is " + sum/arr.length );

}

}

**Lab Exercise No:**21

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to construct two matrices and display the sum of those.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A program to add two matrices

\*

\* Created Aniket Wattamwar

\*/

**public** **class** AddTwoMatrices {

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

// The matrices can be made by taking input from the user using scanner and number of rows and columns

//creating two matrices

**int** mat1[][]= { {1,1,1},

{1,1,1},

{1,1,1}

};

**int** mat2[][]= { {1,1,1},

{1,1,1},

{1,1,77}

};

//storing the result in sum array

**int** sum[][]=**new** **int**[3][3];

//adding the two matrices

**for** (**int** c = 0; c < 3; c++)

**for** (**int** d = 0; d < 3; d++)

sum[c][d] = mat1[c][d] + mat2[c][d];

//displaying the result

**for** (**int** c = 0; c < 3; c++)

{

**for** (**int** d = 0; d < 3; d++)

System.***out***.print(sum[c][d] + "\t");

System.***out***.println();

}

}

}

**Lab Exercise No:**22

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to display the square of the elements of a two dimensional array.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A program to square the elements of a 2d array

\*

\* Created Aniket Wattamwar

\*/

**public** **class** Square2DArray {

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

// The matrices can be made by taking input from the user using scanner and number of rows and columns

//creating a matrix

**int** mat1[][]= { {1,1,1},

{1,1,1},

{1,1,11}

};

//storing the result in sum array

**int** sq[][]=**new** **int**[3][3];

//squaring of the elements of the matrix

**for** (**int** c = 0; c < 3; c++)

**for** (**int** d = 0; d < 3; d++)

sq[c][d] = mat1[c][d] \* mat1[c][d];

//displaying the result

**for** (**int** c = 0; c < 3; c++)

{

**for** (**int** d = 0; d < 3; d++)

System.***out***.print(sq[c][d] + "\t");

System.***out***.println();

}

}

}

**Lab Exercise No:**23

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to construct an array with 10 elements and to find the number of*

*occurrences of each element in the Array.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A program to calculate the occurrences of each element in an array

\*

\* Created Aniket Wattamwar

\*/

**public** **class** Frequency {

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

//creating an array

**int** arr[] = {2,2,2,4,4,5,5,5,6};

**int** count ;

**int** counted = -1;//this is used so that we wont count the same element again

//frequency array

**int**[] frequency = **new** **int**[arr.length];

**for**(**int** i=0;i< arr.length;i++) { //iterating through the array

count = 1; //counter reset to 1

**for**(**int** j = i+1;j<arr.length;j++) {

**if**(arr[i]==arr[j]) { //check whether the element is same or not

count++;

frequency[j]=counted;

}

}

**if**(frequency[i]!=counted) { //add the count to the frequency array

frequency[i]=count;

}

}

//displaying the frequency array

**for**(**int** i =0;i<frequency.length;i++) {

**if**(frequency[i]!=counted) {

System.***out***.print(frequency[i] + ",");

}

}

}

}

**Lab Exercise No:**24

**Exercise Objective(s):***Overloading*

**Exercise:***Create a class called shape with the following methods*

1. *area*
2. *perimeter*

*Overload the area and perimeter method to calculate for both square and rectangle.*

*Create a main class and invoke the area method to calculate the area of the square and*

*rectangle. Also invoke the perimeter method to calculate the perimeter of the square*

*and rectangle.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A program to show how method overloading works using shape class and object as Square and Rectangle

\*

\* Created Aniket Wattamwar

\*/

**public** **class** Shape {

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

Shape square = **new** Shape();

**int** areaSquare = square.area(5);

System.***out***.println("The area of the square is "+ areaSquare);

**int** perimeterSquare = square.perimeter(5);

System.***out***.println("The perimeter of the square is "+ perimeterSquare);

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*");

Shape rectangle = **new** Shape();

**int** areaRectangle = rectangle.area(5,6);

System.***out***.println("The area of the rsectangle is "+ areaRectangle);

**int** perimeterRectangle = rectangle.perimeter(5,6);

System.***out***.println("The perimeter of the rectangle is "+ perimeterRectangle);

}

**public** **int** area(**int** side) {

**return** side\*side;

}

**public** **int** area(**int** length, **int** width) {

**return** length\*width;

}

**public** **int** perimeter(**int** side) {

**return** 4\*side;

}

**public** **int** perimeter(**int** length,**int** width) {

**return** 2\*(length+width);

}

}

**Lab Exercise No:**25

**Exercise Objective(s):***Overloading*

**Exercise:** *Create a class called employee with the following data members*

1. *empName*
2. *empId*
3. *empAge*
4. *empdesgn*
5. *empLocation*
6. *empExpInYrs*

*All these data members should be initialized using constructors. Use constructor overloading*

*and demonstrate by creating different employee objects with*

1. *Employee name alone*
2. *Employee name and id*
3. *Employee name, id and age*
4. *Employee name, id and designation*
5. *Employee name, id, age and designation*
6. *Employee name, id, age and location*
7. *Employee name, id, age and experience*
8. *Employee name, id, designation and experience*
9. *Employee name, id, designation, location and experience*
10. *Employee name, id, age, designation, location and experience*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**package** com.hsbc.solutions;

/\*

\* A program to show constructor overloading

\*

\* Created Aniket Wattamwar

\*/

**public** **class** Employee {

**private** String empName;

**private** **int** empId;

**private** **int** empAge;

**private** String empDesgn;

**private** String empLocation;

**private** **int** empExpInYears;

**public** Employee(String empName, **int** empId, **int** empAge, String empDesgn, String empLocation, **int** empExpInYears) {

**super**();

**this**.empName = empName;

**this**.empId = empId;

**this**.empAge = empAge;

**this**.empDesgn = empDesgn;

**this**.empLocation = empLocation;

**this**.empExpInYears = empExpInYears;

}

**public** Employee(String empName, **int** empId, String empDesgn, String empLocation, **int** empExpInYears) {

**super**();

**this**.empName = empName;

**this**.empId = empId;

**this**.empDesgn = empDesgn;

**this**.empLocation = empLocation;

**this**.empExpInYears = empExpInYears;

}

**public** Employee(String empName, **int** empId, String empDesgn, **int** empExpInYears) {

**super**();

**this**.empName = empName;

**this**.empId = empId;

**this**.empDesgn = empDesgn;

**this**.empExpInYears = empExpInYears;

}

**public** Employee(String empName, **int** empId, **int** empAge, **int** empExpInYears) {

**super**();

**this**.empName = empName;

**this**.empId = empId;

**this**.empAge = empAge;

**this**.empExpInYears = empExpInYears;

}

**public** Employee(String empName, **int** empId, String empDesgn) {

**super**();

**this**.empName = empName;

**this**.empId = empId;

**this**.empDesgn = empDesgn;

}

**public** Employee(String empName, **int** empId, **int** empAge) {

**super**();

**this**.empName = empName;

**this**.empId = empId;

**this**.empAge = empAge;

}

**public** Employee(String empName, **int** empId) {

**super**();

**this**.empName = empName;

**this**.empId = empId;

}

**public** Employee(String empName) {

**super**();

**this**.empName = empName;

}

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

Employee e1 = **new** Employee("Aniket");

Employee e2 = **new** Employee("Rahul",54321);

Employee e3 = **new** Employee("Genie",12345,1000);

Employee e4 = **new** Employee("Smita",9999,"software Engineer");

Employee e5 = **new** Employee("Mr. Bose", 9092,45,22);

Employee e6 = **new** Employee("Ms. Khurana",12211, "Manager", 17);

Employee e7 = **new** Employee("Dr. Arora",1001, " Senior Manager", "Pune ", 28);

Employee e8 = **new** Employee("Mr. XYZ",007, 45, " CEO", "London", 35);

}

}

**Lab Exercise No:**26

**Exercise Objective(s):***Overloading*

**Exercise:***Create a class called Calculator which has 4 different methods add, diff, mul and div which*

*accepts two numbers as parameters. Overload the methods such that the parameters can be*

*of the following pattern.*

1. *Both are of int data type.*
2. *Both are of double data type.*
3. *First parameter is of int data type and second parameter is of double data type.*
4. *First parameter is of double data type and second parameter is of int data type.*

*Create anobject to access these methods and invoke these methods with different type of*

*numbers and display the result in the corresponding methods.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *Re-use the code from Lab Exercise 14*

**package** com.hsbc.solutions;

/\*

\* A program to show how overloading can be used in calculator application

\*

\* Created Aniket Wattamwar

\*/

**public** **class** CalculatorOverloading {

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

//Taking input two numbers

// float x = Integer.parseInt(args[0]);

// float y = Integer.parseInt(args[1]);

**int** ans = *add*(1,3);

System.***out***.println("The addition of the two numbers is "+ ans);

**double** ansInDouble = *add*(11.4d,10.1d);

System.***out***.println("The addition of the two numbers is(double datatype) "+ ansInDouble);

**int** sub = *subtract*(4,11);

System.***out***.println("The difference of the two numbers is "+ sub);

**double** subInDouble = *subtract*(4,11.555d);

System.***out***.println("The difference of the two numbers is(one double and one int value passed) "+ subInDouble);

**int** divide = *divide*(5,5);

System.***out***.println("After dividing the two number. The result is "+ divide);

**int** mul = *multiply*(6,4);

System.***out***.println("The multiplication of the two numbers is "+ mul);

**double** mulInDouble = *multiply*(6.234d,4);

System.***out***.println("The multiplication of the two numbers is(one is double and one int value passed) "+ mulInDouble);

}

//addition of two numbers

**public** **static** **int** add(**int** x, **int** y) {

**return** x+y;

}

**public** **static** **double** add(**double** x, **double** y) {

**return** x+y;

}

//Subtraction of two numbers

**public** **static** **int** subtract(**int** x, **int** y) {

**return** x-y;

}

**public** **static** **double** subtract(**int** x, **double** y) {

**return** x-y;

}

//Division of two numbers

**public** **static** **int** divide(**int** x, **int** y) {

**return** x/y;

}

//Multiplication of two numbers

**public** **static** **int** multiply(**int** x, **int** y) {

**return** x\*y;

}

//Multiplication of two numbers with double and int as parameter

**public** **static** **double** multiply(**double** x, **int** y) {

**return** x\*y;

}

}

**Lab Exercise No:**27

**Exercise Objective(s):***Initializers*

**Exercise:***Write a class called Computer such that the object of that class should be created only when*

*the class is loaded.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**28

**Exercise Objective(s):***Var-args*

**Exercise:***In the calculator (Lab exercise - 14) program, make the add and diff method to accept var-args*

*and demonstrate.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):***NA*

**package** com.hsbc.solutions;

/\*

\* A program to show how unlimited arguments can be passed

\*

\* Created Aniket Wattamwar

\*/

**public** **class** VarArgs {

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

*add*(4, 13, 15, 11, 27,34);

*sub*(50, 17, 46, 8, 14,55);

}

// ...num will make this function with unlimited arguments.

//addition of 2 number

**public** **static** **void** add(**int** ...num) {

**int** sum = 0;

**for**(**int** i = 0; i < num.length; i++){

sum += num[i];

}

System.***out***.println("Addition of numbers is "+ sum);

}

//subtraction of number

**public** **static** **void** sub(**int** ...num) {

**int** sub = num[0]

**for**(**int** i = 1; i < num.length; i++){

sub -= num[i];

}

System.***out***.println("The difference is "+ sub);

}

}