S. S. V. P. S.'s B. S. Deore College of Engineering, Dhule.

Name : Jadhav Sandhya Anil

Roll No. : 145 Class : SY BTech Batch : B3

Expt No. : 2 Subject : Java.

Expt. :

Date of Performance	Date of Completion	Marks / Grade	Sign. & Date

Title/Aim :

Goal :-

Prerequisite :- Algorithm, Basic of Java.

Software Requirement :- Netbean, JDk 1.8.

Theory

Array :-

An array is collection of elements of similar type that shares a common name.

- In Java all arrays are dynamically allocated.
- Since arrays are object in Java, we can find their length using member length.

Incomplete for -

CKT diag.

Observation

Calculations /Result

Graph

Theory

- A Java array variable can also be declared like other variable with [] after data type.
- The variables in array are ordered and each have in index beginning from 0
- Java array can be also be used as a static field,

a local variable or a method parameter.
- The size of an array must be specified by an int value and not long or short.

- Creating, Initializing and Accessing an Array

i) One - Dimensional Arrays :-

- One dimensional arrays is created using following step
1. Declaring an array
2. Creating memory location
3. Initializing an array.

1] Declaring an array :-
Arrays in Java can be declare in two froms.
i) data_type[ ] arrname ;
or
ii) data-type arrname[ ];

data-type declares the element type of array. The element type of determines the data type of each element that comprises the array. element type for the array determine what type of data the array will hold.

2) Creating memory location:
When an array is declared, only a reference of array is created. To actually create or given memory

to array, you create an array like this:

Syntax:-

arrName = new data-type[size]

Here, type specifies the type of data being allocated,
size, specifies the number of elements in array,
var-name, is the name of array variable that is linked to the array.

That is to use new to allocate an array, you must specify the type and number of ele. to allocate.

Example:- int array[];
array = new int[20];
OR
int array[] = new int[20];

- The element in the array allocated by new will automatically be initialized to zero, ~~false or null~~
- Obtaining an array is a two-step process. First, you must declare a variable of desired array type. Second, you must allocate the memory that will hold the array, using new and ~~d~~ assign it to the array variable. Thus, in Java all array are dynamically allocated.

3] Initialization of an array.

- putting values in array called array initialization.
- This can be done by using array index.

Syntax : arrName[index] = value.

Example : arrName[0] = 10;

arrName[1] = 20;

We can initialize the array at the time of declaration.

Syntax :-

datatype arrName[] = { list of value };

Example :-

int a[] = { 10, 30, 20, 40 }

ii) Two - Dimensional Array.

In Java, multidimensional arrays are actually arrays of arrays.

To declare a multidimentional array variable, specify each additional index using another set of square brackets.

1. Declare in array

Syntax :- datatype arrName[][];

or

datatype[][] arrname;

2. Creating Memory Location.

Syntax :-

arrayName = new datatype[rows][column]

or

datatype arrayName[][] = new datatype[][]

3. Initialization of two dimensional Array

Syntax:-

arrName[rowIndex][colIndex] = value;

Example:-

a[0][1] = 20

a[1][0] = 30

We can initialize the array at the time of declaration.

Syntax:-

data-type arrName[][] = { list of value }

Example:-

int a[3][2] = {10, 30, 20, 40, 50, 70}

Or

int a[3][2] = {{10,30}, {20,40}, {50,70}}

Implementation:-

Source code of the program is in Java.

Conclusion:-

S. S. V. P. S.'s B. S. Deore College of Engineering, Dhule.

Name : Jadhav Sandhya Anil

Roll No. : 145 Class : SY. B.Tech Batch : B3

Expt No. : 3 Subject : Java

Expt. :

Date of Performance	Date of Completion	Marks / Grade	Sign. & Date

Title/Aim :

Goal :- Perform basic Method on String and Math class.

Prerequisite :- Algorithm, Basic of Java

Software Requirement :- Netbeans, JDK 1.8.

Theory / Analysis

String Class :-

The String class represents character String String can be Create Using String class. String class create immutable string that mean once we create object of string class we can not change its value ; but the variable declared as String reference can be Changed to point at some other String object.

Incomplete for -

CKT diag.

Observation

Calculations /Result

Graph

Theory

Create String Using String class.

Syntax :For create a String object

Java String is created by using keyword "new".

String var-name = new String ("Hello");

or

String var-name = value ;

Java string literal is created by using double quotes.

Java String pool

Java string pool refers to collection of string which are stored in heap memory. In this whenever a new object is created, String pool first check whether the object is already present in the pool or not. If it's present, then same reference is returned to the variable else new object will be created in the string pool and the respective reference will be returned.

Methods of Java String class

1. String toUppercase() : convert all the character in invoking string to uppercase.

Example :- String S1 = "hello";

String S2 = S1.toUpperCase();

output : "HELLO"

2. String toLowerCase() : convert all the characters in invoking string to lowercase.

3. replace() : Replaces all the accurrences of oldchar with, newchar in the invoking string.

Syntax :- String replace (char oldchar, char newchar)

4) trim() : Removes whit spaces at the beinning and at the end of the involing string
Syntax :- String trim()

5) length() : Return the length of the invoking string i.e number of character from involing string
Syntax :- int length()

6) concat() : Concatenates str with involing string
Syntax : String concat(String str)

7) charAt() : Return the character at the position Specified by the parameter index from the involing String.
Syntax : char charAt (int index)

8) indexOf() : Return the index position of the character specified in parameter ch from the involing String.
Syntax : int indexOf (char ch) int

9) subString() : Returns sub string from the involing string from being Index up to end.
The second from return sub string from the involing string start at being Index up to but not including character at end Index.
Syntax :- String Substring (int beginIndex)
String substring (int beginIndex, int endIndex)

10) equals() :- Return true if invoking string is equals to the String passed in parameter Str, otherwise False
Syntax :- boolean equals(Object Str)

11) equalsIgnorCase() :- Returns true if involing string is equals to the String passed in parameter Str ignoring the cases of characters, otherwise return False.
Syntax :- boolean equalsIgnoreCase(String Str)

12) CompareTo() :- Compares involing string with the String passed in parameter Str and interpreter the result as follow:
Syntax :- int CompareTo(String Str)

2) Math class in Java

The java.lang.Math class contains Methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and and trigonometric function.

Basic Math Methods

1) Math.abs() - it will return the Absolute value of given value
2) Math.max() - It returns the largest of two values.
3) Math.min() - It is used to return the Smallest of 2 values.
4) Math.round() - It is used to return round of the decimal numbers to the nearest value.

5) Math.sqrt() : It is used to return the square root of a number.

6) Math.log() : It returns the natural logarithm of a double value.

7) Math.sin() : It is used to return the trigonometric Sine value of a Given double value.

8) Math.incrementExact() : It returns the argument increment by one.

9) Math.exp() : It returns E raised to the power OF a double value, where E is Eulers number and it is approximately equal to 2.71828.

Implementation :-

Source code of the program is in Java

Conclusion :-

S. S. V. P. S.'s B. S. Deore College of Engineering, Dhule.

Name : Jadhav Sandhya Anil

Roll No. : 145 Class : SY. B.Tech Batch : B3

Expt No. : 4 Subject : Java

Expt. :

Date of Performance	Date of Completion	Marks / Grade	Sign. & Date

Title/Aim :

Goal :

prerequisite :- Algorithm, Basic of Java

Software requirement :- Netbeans, JDK 1.8

Theory / Analysis

1) Inheritance :-

Inheritance is the process by which object of one class can acquire properties of object of another class.

Inheritance is the mechanism of deriving a new class from an old class. The old class is called as base class or parent class or super class & new class is called derived class or sub class.

Incomplete for -

CKT diag.

Observation

Calculations /Result

Graph

Theory

~~class~~ How to use Inheritance in Java.

The keyword used for inheritance is extends.

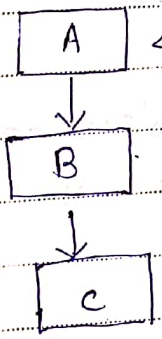
Syntax :

```
class derived-class extends base-class
{
    // method and fields.
}
```

Type of Inheritance

i) Single inheritance :- When there is only one base class having one derived class then such inheritance is called single inheritance.

ii) Multilevel Inheritance :- In this type of inheritance derived class act as base class for another class.



← Base class

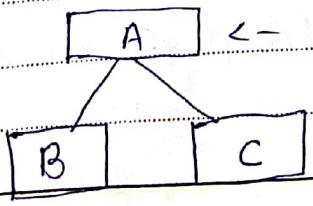
"B is derived from A"

← Derived class

"C is derived from B"

← Derived class

iii) Hierarchical Inheritance :- When there is one base class having multiple derived class, such inheritance is called Hierarchical inheritance.



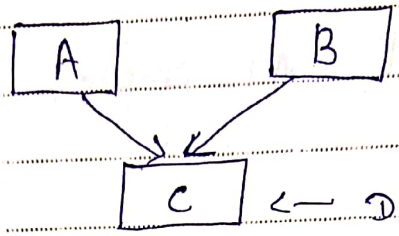
← Base class.

Derived class →

← Derived class

iv) Multiple inheritance :-

When there are more than one base classes and having single derived class, such inheritance is called : Multiple inheritance.



" C is derived from A and B"

In Inheritance used two keyword.

i) extends :- The keyword extends signifies that properties of the 'superclassName' are extended by 'subclassName'. Subclass now contain its own instance variables and method as well as those of Superclass.

ii) Super :- The super keyword used to access variable and method of superclass into a subclass.

Syntax :-

```
Super.variablename;
Super.methodname();
```

2) Polymorphism

Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance.

Polymorphism uses those methods to perform different task. This allow us to perform a single action in different ways.

i) Compile time polymorphis

- polymorphism that is resolved during compiler time is known as static polymorphism.
Method overloading is an example of static polymorphism.

- Method Overloading :-

This allows us to have more than one method having the same name, if the parameters of methods are different in number, sequence and data type of parameter.

Method overloading is one of the way Java support static polymorphism. Return type does not considered of the time of method overloading.

- Method overloading is done only in same scope (same class)
- In method overloading compiler resolves call to compile time hence it is also known as compile time polymorphism.

Example :-

```
class Demo
{
    int add(int a, int b)
    {   return(a+b);
    }
    float add(float x, float y)
    {   return (x+y);
    }
```

ii) Runtime Polymorphism

It is also known as Dynamic Method dispatch.

Dynamic polymorphism is a process in which a call to an overridden method is resolved at runtime, that's why it is called runtime polymorphism.

Method overriding is a example of Dynamic polymorphism.

- Method overriding

When method in subclass has same name and same type signature as method in its Superclass. then method in Subclass is said to override the method in the Super class.

- When an overridden method is called, then method defined in the Subclass is invoked and execute instance of the method defined in Superclass. The Method define in super class will be hidden This is know as Method overriding.

Implementation :-

Source Code of the program is in Java

Conclusion :-

S. S. V. P. S.'s B. S. Deore College of Engineering, Dhule.

Name : Jadhav Sandhya Anil

Roll No. : 145 Class : SY. B.Tech Batch : B3

Expt No. : 7 Subject : Java

Expt. :

Date of Performance	Date of Completion	Marks / Grade	Sign. & Date

Title/Aim :

Goal :

Prerequisite : Algorithm, Basic of Java

Software Requirement: Algorithm,

Theory /Analysis :

Exception - Handling in Java

An exception is an abstract condition that is caused by a run-time error in the program. Exception-Handling enables you to create application that can handle exception. In many cases, handling an exception allow a program to continue executing as if no problem had been encountered.

Incomplete for -

CKT diag.

Observation

Calculations /Result

Graph

Theory

when Java Enterpreter encounter an error such as dividing an integer by zero, it create an exception object and throws it. (i.e. inform than an error has occured)

If exception object is not caught and handle proper the interpreter will display an error msg and terminate the program.

If we want the program to continue with the execution of remaning code then we should try to catch the exception object thrown by the error condition and then display an appropriate message for tasking correct action. Thus task is known as "Exception Handling".

In Java exception Handling is managed via Five keywords.

1. try :- A try keyword is used to indicate a block of code that is likely to cause an error.
2. catch :- The Exception thrown by the try block and handle it appropriatly. The catch block is added immediate after try block.
3. throw :- To manually throw an exception, use throw.
4. throws :- Any exception that is thrown out of a method must be specified as such by a throws clause.
5. Finally :- Any code that absolutely must be executed before method return is put in a finally block.

General from of an exception - Handling block

```
try {
    // block of code that cause an exception
}
catch (ExceptionType exOb) {
```

// exception handle for ExceptionType
}
Finally {
// block of code to be executed before
try block end.
}

Exception Type

All exception types are subclasses of the build-in class Throwable.

The Exception class does not define any method of its own. It does, of course, inherit those method provided by Throwable.

There is an important subclass of Exception, is called RuntimeException.

i) checked Exception :- The classes which direcly inherit Throwable class except RuntimeException and Error are know as checked Exception. Checked exception are checked at compile time. e.g :- IOException

ii) Unchecked Exception :- The classes which inherit RuntimeException are known as unchecked exceptions. unchecked exception are ~~not~~ checked at Runtime.

iii) Error :- Error is irrecoverable

eg :- OutOfMemoryError, VirtualMachineError

// Common scenarios of Java Exception.

Java Unchecked Runtime Exception subclasses.

i) ArithmaticException :- Arithmatic error such as divide by zero

ii) ArrayIndexOutOfBoundException :- Array index out of Bound.

iii) IndexOutOfBoundException :- Some type of index is out of Bound.

iv) NegativeArraySizeException :- Array Create with a Negative Size

~~v) NullPointerExcept~~

v) NullPointerException :- Invalid use of a null reference.

vi) NumberFormatException :- Invalid Conversion of a String to a numeric Format.

vii) StringIndexOutOfBound :- Attempt to index outside the bounds of a String.

viii) UnsupportedOperationException :- An unsupported aperation was encountered.

Java checked Exception.

i) ClassNotFoundException :- class not found.

ii) InstantiationException :- Attempt to Create an object of an abstract class

iii) InterruptedException :- one thread has been interrupted by other thread.

iv) NoSuchMethodException :- A requested method does not exist.

v) NoSuchFieldException :- A requested field does not exist.

What is Exception Handling ?

Exception Handling is a mechanism to handle runtime error such as classNotFoundException, IOException, RemoteException etc.

Advantage of Exception Handling.

Exception Handling is to maintain the normal flow of the application. An exception normally disrupts the normal flow of the application that is why we use exception Handling.

Implementation :- Source code of the program is in Java

Conclusion :-

Execute the Java program Some time we known in any method occure Exception then we can use in Exception Handling.