****

**SIX WEEKS SUMMER TRAINING REPORT**

on

***(CORE JAVA)***

Submitted by

**(VEMPALI B N V D SIVASANTHOSH)**

**Registration NO =11705138**

**Programme Name:CSE**

Under the Guidance of

**SARVESH ARGRAWAL ( INTERNSHALA)**

**School of Computer Science & Engineering**

**Lovely Professional University, Phagwara**

(June-July, 2019)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Annexure-III |
|  |  |  | **DECLARATION** | | | |  |  |

I hereby declare that I have completed six weeks summer training at INTERNSHALA from

June 1 to July 13 Under the Guidance of Sarvesh agarwal (INTERNSHALA ONLINE TRAININGS). I have declare that I have worked with full dedication during these six

weeks of training and my learning outcomes fulfill the requirements of training for the award

of degree of COMPUTER SCIENCE AND ENGINEERING Lovely Professional University, Phagwara.

(Signature of student)

Name of Student VEMPALI B N V D SIVASANTHOSH

Registration no: 11705172

Date: 29-07-2019

**Acknowledgement :**

First of all, I would like to express my gratitude to SARVESH AGARWALl(INTERNSHALA) to enabling me to complete this report

on **CORE JAVA.**

Successfully completion of any type of project requires help from a number of persons. I’ve also taken help

from different people for the preparation of this report. Now, there is a little effort to show my deep gratitude

to that helpful person.

I convey my sincere gratitude  **TO SARVESH AGARWAL( Internshala.)** WhO helped me to move through an right path and

direction and proper guidance this study would have been a little success. In every phase of the project

his supervision and guidance shaped this report to be completed perfectly.

I would also like to thank my friends in my Batch to give a vivid idea about the CORE JAVA**.**

**CERTIFICATE OF COMPLETION**

****

**CONTENT**

1. Acknowledgement...........................................................................................................i
2. Introduction.....................................................................................................................1
3. Organisation....................................................................................................................2
4. Basics Of Java.................................................................................................................2
   1. Overview Of Java...................................................................................................2
   2. Features Of Java.....................................................................................................3
   3. Java ClassPath Setting............................................................................................4
   4. Steps For Compiling And Executing Programs..........................................................4
   5. Difference Between JDK,JRE And JVM...................................................................5
   6. Object And Class........................................................................................................5
   7. Data Types..................................................................................................................6
   8. Variables.....................................................................................................................7
   9. Operators And Its Types..............................................................................................7
5. Programming Concepts......................................................................................................7
   1. Structure Of Java Program..........................................................................................7
   2. Main Method...............................................................................................................8
   3. Decision Making Statement........................................................................................8
   4. Looping Statement......................................................................................................8
   5. Wrapper Classes..........................................................................................................9
   6. Access Modifiers.........................................................................................................9
6. Java Useful Keywords........................................................................................................9
   1. Final.............................................................................................................................9
   2. Static ...........................................................................................................................9
   3. This..............................................................................................................................10
   4. Super............................................................................................................................10
   5. Synchronized...............................................................................................................10
7. Java Classes, Blocks And Constructors...............................................................................11
   1. Abstract Classes............................................................................................................11
   2. Constructors..................................................................................................................11
   3. Constructors And Its Types...........................................................................................11
   4. Static Block In Java.......................................................................................................11
   5. Relationship In Java......................................................................................................11
8. Java Object Oriented........................................................................................................12

8.1 . Inheritance.................................................................................................................12

8.2. Method Overloading..................................................................................................13

8.3. Method Overriding....................................................................................................13

8.4. Interface.....................................................................................................................14

8.5. Abstraction................................................................................................................14

8.6. Encapsulation............................................................................................................15

8.7. Polymorphism...........................................................................................................15

9. Java Advanced..................................................................................................................16

9.1. Package......................................................................................................................16

9.2. Exception Handling...................................................................................................16

9.3. Multithreading...........................................................................................................17

9.4. String..........................................................................................................................17

9.5. Collection...................................................................................................................18

10. Projects Undertaken........................................................................................................19

Learning outcomes ……………………………………………………………………………………………………….20

Conclusions.......................................................................................................................21

References.........................................................................................................................22

**INTRODUCTION**

Java is one of the programming language or technology used for developing web applications. Java language developed at SUN Micro Systems in the year 1995 under the guidance of James Gosling and there team. Originally SUN Micro Systems is one of the Academic university (Standford University Network)

Whatever the software developed in the year 1990, SUN Micro Systems has released on the name of oak, which is original name of java (scientifically oak is one of the tree name). The OAK has taken 18 months to develop.The oak is unable to fulfill all requirements of the industry. So James Gosling again reviews this oak and released with the name of java in the year 1995. Scientifically java is one of the coffee seed name.

**Java divided into three categories, they are**

* J2SE (Java 2 Standard Edition)
* J2EE (Java 2 Enterprise Edition)
* J2ME (Java 2 Micro or Mobile Edition)

**J2SE**

J2SE is used for developing client side applications.

**J2EE**

J2EE is used for developing server side applications.

**J2ME**

J2ME is used for developing mobile or wireless application by making use of a predefined protocol called WAP (wireless Access / Application protocol).

## ORGANISATION

## HP the largest IT company in the world, through its partner offers high quality Courses for 4 & 6 weeks. Few Courses in Summer Training Program are Programming Techniques using 'C', PHP, VLSI, VHDL & PCB Design, Data Structure using C Language, Core Java, J2EE - Hibernate with Spring Framework, J2EE - Struts with Hibernate Framework, Android, ASP.NET With C#, Embedded & Robotics-Basics & Advanced, ARM, Linux Administration with Scripting, Networking concepts with security.

## BASICS OF JAVA

## Overview Of Java

Java is a platform independent, more powerful, secure, high performance, multithreaded programming language. Here we discuss some points related to java.

## Define JRE

The **Java Runtime Environment (JRE)** is part of the Java Development Kit (JDK). It contains set of libraries and tools for developing java application. The Java Runtime Environment provides the minimum requirements for executing a Java application.

## Define JVM

**JVM** is set of programs developed by sun Micro System and supplied as a part of jdk for reading line by line of byte code and it converts into native understanding form of operating system. Java language is one of the compiled and interpreted programming language.

## Garbage Collector

**Garbage Collector** is the system Java program which runs in the background along with regular Java program to collect un-Referenced (unused) memory space for improving the performance of our applications.

## Define an API

An **API (Application Programming Interface)** is a collection of packages, a package is the collection of classes, interfaces and sub-packages. A sub-package is a collection of classes interfaces and sub sub packages etc.

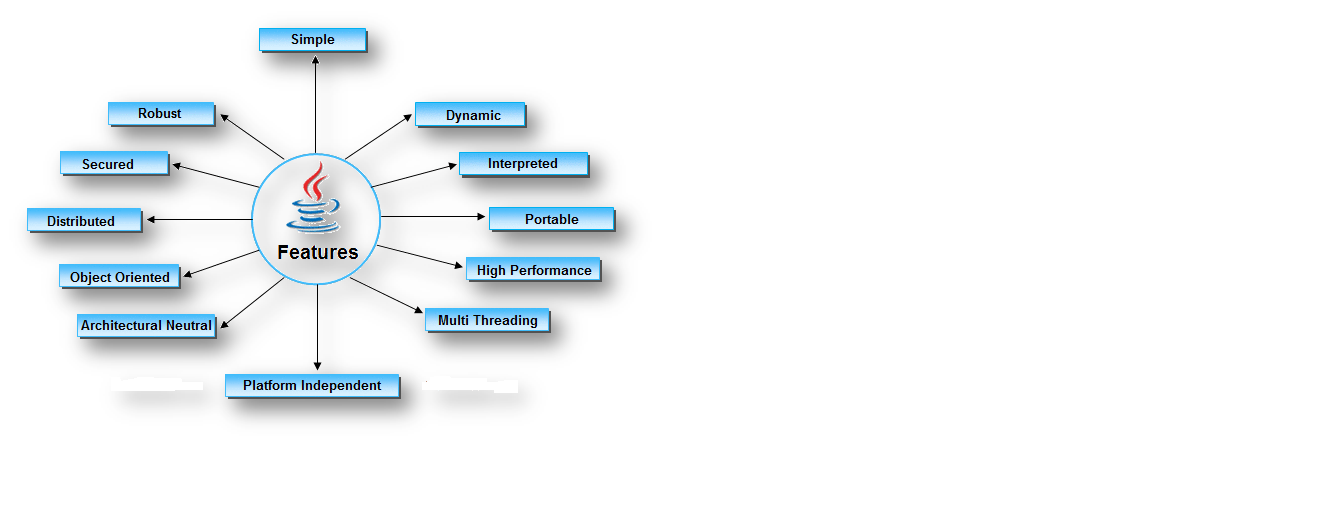
Java programming is containing user friendly syntax so that we can develop effective application s. in other words if any language is providing user friendly syntax, we can develop error free applications.

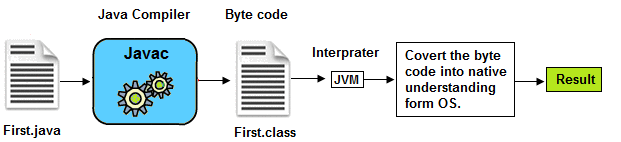
## Definition of JIT

JIT is the set of programs developed by SUN Micro System and added as a part of JVM, to speed up the interpretation phase

**Features Of JAVA**

Features of a language are nothing but the set of services or facilities provided by the language vendors to the industry programmers. Some important features are;





## Java Classpath Setting

### Path Variable

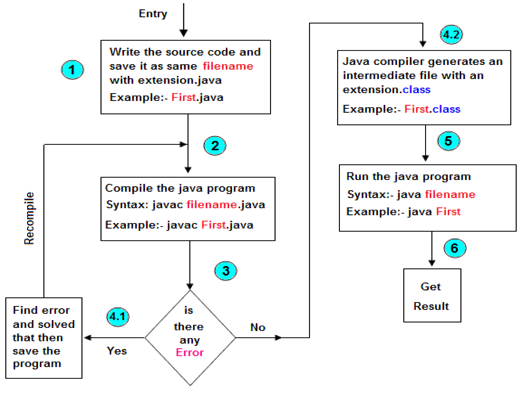
Path variable is set for providing path for all java tools like java, javac, javap, javah, jar, appletviewer which are use in java programming. These all tools are available in **bin** folders so we set path upto bin folders.

### Classpath Variable

**Classpath** variable is set for providing path for predefined java classes which is used in our application. All classes are available in **lib/rt.jar** so we set classpath upto lib/rt.jar.

**Steps For CompilingAnd Executing Programs**

The following sequence of steps represented in the diagram use compiling the java program and executing the java programs.



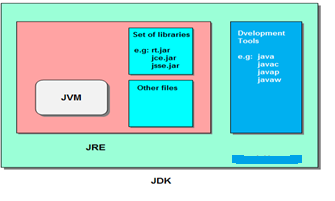
## Difference Between JDK,JRE AND JVM

Jvm, Jre, Jdk these all the backbone of java language. Each components have separate works. Jdk and Jre physically exists but Jvm are abstract machine it means it not physically exists.

**JVM :** JVM (Java Virtual Machine) is a software. It is a specification that provides runtime environment in which java bytecode can be executed. It not physically exists.JVMs are not same for all hardware and software, for example for window os JVM is different and for Linux VJM is different. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent.

**JRE :** The Java Runtime Environment (JRE) is part of the Java Development Kit (JDK). It contains set of libraries and tools for developing java application. The Java Runtime Environment provides the minimum requirements for executing a Java application. It physically exists. It contains set of libraries + other files that JVM uses at runtime.

**JDK** : The Java Development Kit (JDK) is primary components. It physically exists. It is collection of programming tools and JRE, JVM.



**Object And Class**

Object is the physical as well as logical entity where as class is the only logical entity.**Class:** Class is a blue print which is containing only list of variables and method and no memory is allocated for them. A class is a group of objects that has common properties.

A class in java contains:

* Data Member
* Method
* Constructor
* Block
* Class and Interface

**Object:** Object is a instance of class, object has state and behaviors.An Object in java has three characteristics:

* State
* Behavior
* Identity

**State:**Represents data (value) of an object.

**Behavior:**Represents the behavior (functionality) of an object such as deposit, withdraw etc.

**Identity:**Object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. But,it is used internally by the JVM to identify each object uniquely.

Class is also can be used to achieve user defined data types.

**Data Types**

**Datatype** is a spacial keyword used to allocate sufficient memory space for the data, in other words Data type is used for representing the data in main memory (RAM) of the computer.

In general every programming language is containing three categories of data types. They are

* Fundamental or primitive data types
* Derived data types
* User defined data types.

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Default Value** | **Default size** |
| boolean | false | 1 bit |
| char | '\u0000' | 2 byte |
| byte | 0 | 1 byte |
| short | 0 | 2 byte |
| int | 0 | 4 byte |
| long | 0L | 8 byte |
| float | 0.0f | 4 byte |
| double | 0.0d | 8 byte |

## Variable

**Variable** is an identifier which holds data or another one variable is an identifier whose value can be changed at the execution time of program. Variable is an identifier which can be used to identify input data in a program.

**Operators And Its Types**

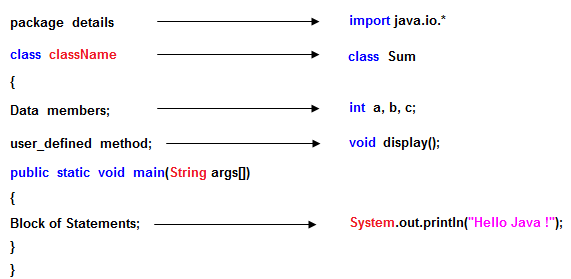
**Operator** is a special symbol that tells the compiler to perform specific mathematical or logical Operation. Java supports following lists of operators.

* Arithmetic Operators
* Relational Operators
* Logical Operators
* Bitwise Operators
* Assignment Operators
* Ternary or Conditional Operators

**PROGRAMMING CONCEPTS**

**Structure Of Java Program**

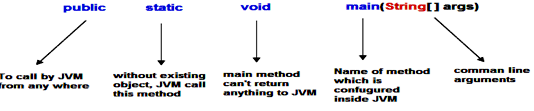
Structure of a java program is the standard format released by Language developer to the Industry programmer. Sun Micro System has prescribed the following structure for the java programmers for developing java application.



* A **package** is a collection of classes, interfaces and sub-packages. A sub package contains collection of classes, interfaces and sub-sub packages etc. java.lang.\*; package is imported by default and this package is known as default package.
* **Class** is keyword used for developing user defined data type and every java program must start with a concept of class.

**Main( ) Method**

**main()** method is starting execution block of a java program or any java program start their execution from main method. If any class contain main() method known as main class.



**Decision Making Statement**

**Decision making statement** statements is also called selection statement. That is depending on the condition block need to be executed or not which is decided by condition. If the condition is "true" statement block will be executed, if condition is "false" then statement block will not be executed. In java there are three types of decision making statement.

* if
* if-else
* switch

**Looping Statement**

## for loop

## While loop

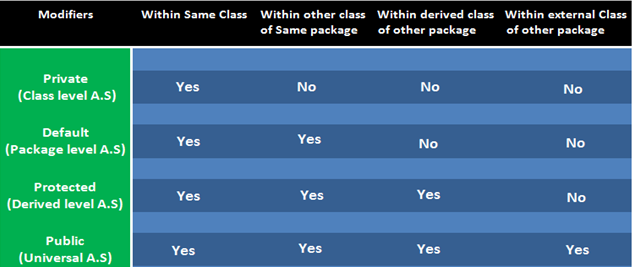
## do-while

## Wrapper Classes

For each and every fundamental data type there exist a pre-defined class, Such predefined class is known as wrapper class. The purpose of wrapper class is to convert numeric string data into numerical or fundamental data.

## Access modifiers:

## **Access modifiers** are those which are applied before data members or methods of a class. These are used to where to access and where not to access the data members or methods. In java programming we have four access modifiers they are



**JAVA USEFUL KEYWORDS**

**Final keyword**

In java language final keyword can be used in following way.

* Final at variable level
* Final at method level
* Final at class level

**Static keyword**

The **static keyword** is used in java mainly for memory management. Static keyword are used with variables, methods, blocks and nested class. Static is a keyword that are used for share the same variable or method of a given class. This is used for a constant variable or a method that is the same for every instance of a class. The main method of a class is generally labeled static.

**In java language static keyword can be used for following**

1. variable (also known as class variable)
2. method (also known as class method)
3. block
4. nested class

## This keyword

**this** is a reference variable that refers to the current object. It is a keyword in java language represents current class object

**"this"** keyword can be use in two ways.

* this . (this dot)
* this() (this off)

**Super keyword**

**Super** keyword in java is a reference variable that is used to refer parent class object. **Super** is an implicit keyword create by JVM and supply each and every java program for performing important role in three places.

* At variable level
* At method level
* At constructor level

## Synchronized Keyword

## Synchronized Keyword is used for when we want to allow only one thread at a time then use Synchronized modifier. If a method or block declared as a Synchronized then at a time only one thread is allowed to operate on the given object.

**JAVA CLASSES, BLOCKS AND CONSTRUCTORS**

**Abstract Classes**

We know that every java program must start with a concept of class that is without classes concept there is no java program perfect.In java programming we have two types of classes they are

1. Concrete class
2. Abstract class

## Constructor And Its Types

A **constructor** is a special member method which will be called implicitly (automatically) by the JVM whenever an object is created for placing user or programmer defined values in place of default values. In a single word constructor is a special member method which will be called automatically whenever object is created.

The purpose of constructor is to initialize an object called object initialization. Constructors are mainly create for initializing the object. Initialization is a process of assigning user defined values at the time of allocation of memory space.

**Types of constructors**

Based on creating objects in Java constructor are classified in two types. They are

* Default or no argument Constructor
* Parameterized constructor.

## Static Block in Java

**Static block** is a set of statements, which will be executed by the JVM before execution of main method. At the time of class loading if we want to perform any activity we have to define that activity inside static block because static block execute at the time of class loading.

In a class we can take any number of static block but all these static block will be execute from top to bottom.

**Relationship in Java**

Type of relationship always makes to understand how to reuse the feature from one class to another class. In java programming we have two types of relationship they are.

* Is-A Relationship
* Has-A Relationship

**JAVA OBJECT ORIENTED**

## Inheritance

The process of obtaining the data members and methods from one class to another class is known as **inheritance**. It is one of the fundamental features of object-oriented programming.

A class that is declared with abstract keyword, is known as **abstract class**. An abstract class is one which is containing some defined method and some undefined method. In java programming undefined methods are known as un-Implemented or abstract method.The process of obtaining the data members and methods from one class to another class is known as **inheritance**. It is one of the fundamental features of object-oriented programming.

**Types of Inheritance**

* Single inheritance
* Multiple inheritance
* Hierarchical inheritance
* Multilevel inheritance
* Hybrid inheritance

**Why use Inheritance ?**

* For Method Overriding (used for Runtime Polymorphism).
* It's main uses are to enable polymorphism and to be able to reuse code for different classes by putting it in a common super class
* For code Re-usability

## Method Overloading

Whenever same method name is exiting multiple times in the same class with different number of parameter or different order of parameters or different types of parameters is known as **method overloading**.

## Why method Overloading ?

Suppose we have to perform addition of given number but there can be any number of arguments, if we write method such as a(int, int)for two arguments, b(int, int, int) for three arguments then it is very difficult for you and other programmer to understand purpose or behaviors of method they can not identify purpose of method. So we use method overloading to easily figure out the program. For example above two methods we can write sum(int, int) and sum(int, int, int) using method overloading concept.

**Different ways to overload the method**

There are two ways to overload the method in java

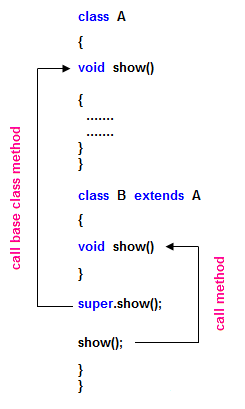
* By changing number of arguments or parameters
* By changing the data type
* By changing the order of arguments.

## Method Overriding

Whenever same method name is existing in both base class and derived class with same types of parameters or same order of parameters is known as **method Overriding**.

**Advantage of Java Method Overriding**

* Method Overriding is used to provide specific implementation of a method that is already provided by its super class.
* Method Overriding is used for Runtime Polymorphism



**Interface**

**Interface** is similar to class which is collection of public static final variables (constants) and abstract methods.The interface is a mechanism to achieve fully abstraction in java. There can be only abstract methods in the interface. It is used to achieve fully abstraction and multiple inheritance in Java.

**Why we use Interface ?**

* It is used to achieve fully abstraction.
* By using Interface, you can achieve multiple inheritance in java.

## When we use abstract and when Interface

If we do not know about any things about implementation just we have requirement specification then we should be go for **Interface**

If we are talking about implementation but not completely (partially implemented) then we should be go for **abstract**

## Abstraction

**Abstraction** is the concept of exposing only the required essential characteristics and behavior with respect to a context.

Hiding of data is known as **data abstraction**. In object oriented programming language this is implemented automatically while writing the code in the form of class and object.

### Real life example of Abstraction

Abstraction shows only important things to the user and hides the internal details for example when we ride a bike, we only know about how to ride bike but can not know about how it work ? and also we do not know internal functionality of bike.

## Encapsulation

**Encapsulation** is a process of wrapping of data and methods in a single unit is called encapsulation. Encapsulation is achieved in java language by class concept.Combining of state and behavior in a single container is known as encapsulation. In java language encapsulation can be achieve using **class** keyword, state represents declaration of variables on attributes and behavior represents operations in terms of method.

**Benefits of encapsulation**

* Provides abstraction between an object and its clients.
* Protects an object from unwanted access by clients.
* Example: A bank application forbids (restrict) a client to change an Account's balance

## Polymorphism

The process of representing one form in multiple forms is known as **Polymorphism**.Here original form or original method always resides in base class and multiple forms represents overridden method which resides in derived classes.

Polymorphism is not a programming concept but it is one of the principal of OOPs. For many objects oriented programming language polymorphism principle is common but whose implementations are varying from one objects oriented programming language to another object oriented programming language.

Polymorphism principal is divided into two sub principal they are:

* Static or Compile time polymorphism
* Dynamic or Runtime polymorphism

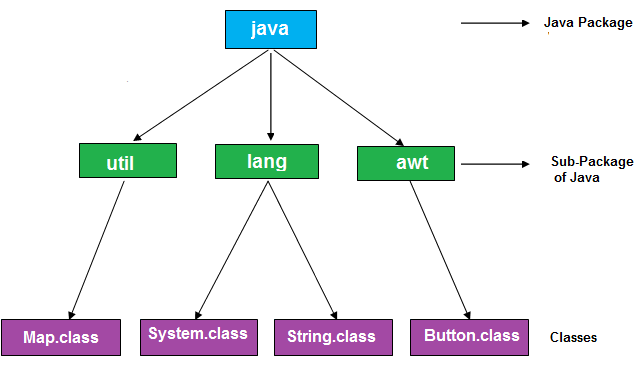
JAVA ADVANCED

## Package

A package is a collection of similar types of classes, interfaces and sub-packages.

## Purpose of package

The purpose of package concept is to provide common classes and interfaces for any program separately. In other words if we want to develop any class or interface which is common for most of the java programs than such common classes and interfaces must be place in a package.



## Exception Handling

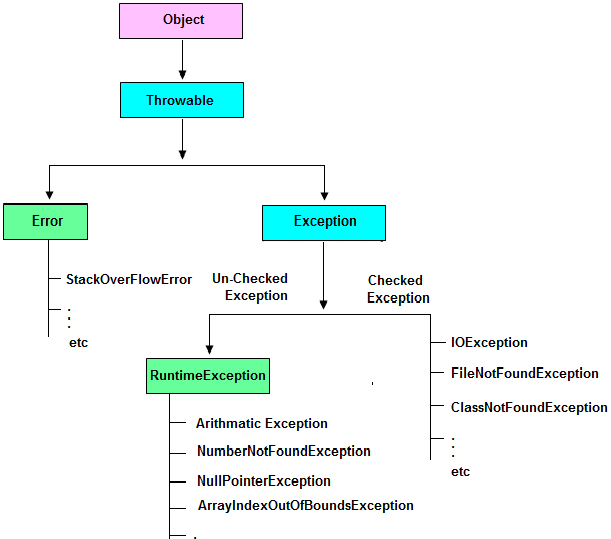
The process of converting system error messages into user friendly error message is known as **Exception handling**. This is one of the powerful feature of Java to handle run time error and maintain normal flow of java application.

An **Exception** is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's Instructions.

**Type of Exception**

* Checked Exception
* Un-Checked Exception

## Hierarchy of Exception classes



**Multithreading**

Multithreading in java is a process of executing multiple threads simultaneously. The aim of multithreading is to achieve the concurrent execution.

**Thread**

Thread is a lightweight components and it is a flow of control. In other words a flow of control is known as thread.State of a thread are classified into five types they are

1. New State
2. Ready State
3. Running State
4. Waiting State
5. Halted or dead State

**String**

String is a sequence of characters enclosed within double quotes (" ") is known as **String.**  
Example: "Java Programming".

In java programming to store the character data we have a fundamental datatype called **char**. Similarly to store the string data and to perform various operation on String data, we have three predefined classes they are:

* String
* StringBuffer
* StringBuilder

# Collection

Collections in java is a framework that provides an architecture to store and manipulate the group of objects.All the operations that you perform on a data such as searching, sorting, insertion, manipulation, deletion etc. can be performed by Java Collections.Java Collection simply means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque etc.) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet etc).

**PROJECTS UNDERTAKEN**

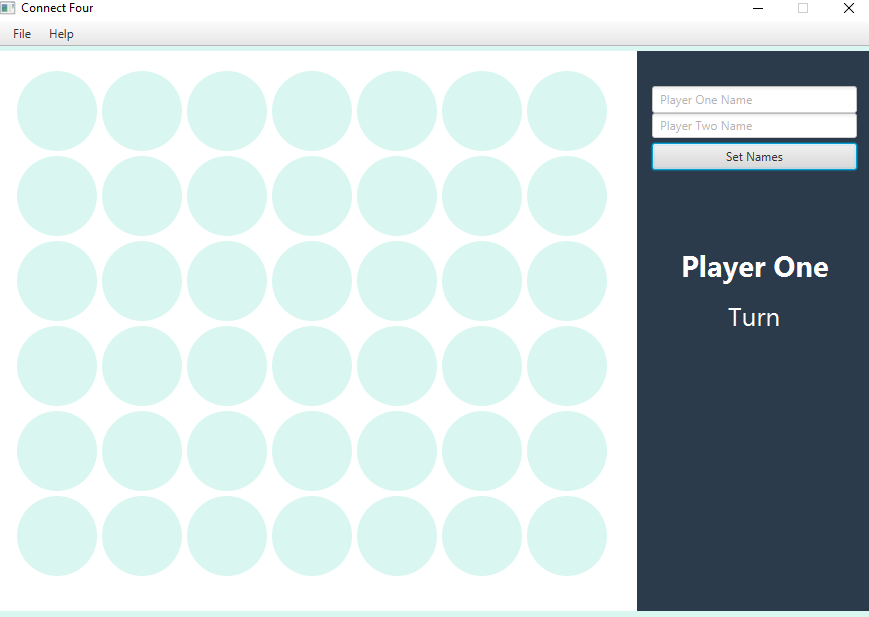
**PROJECT:- Connect four game**

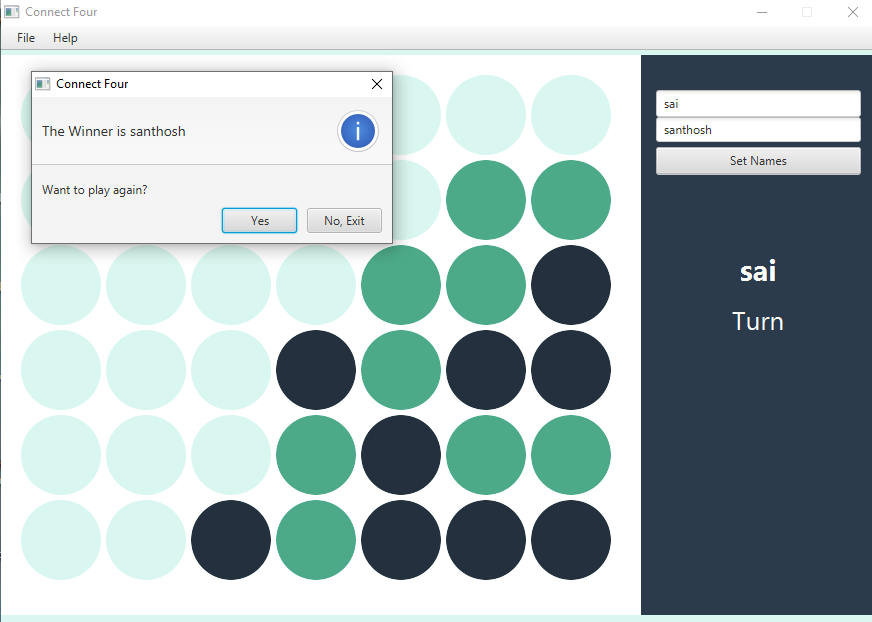
DESCRIPTION:-connect four is an game that two player will play simultaneously in order to

Win the game the player should have to connect the 4 dots in a row in

Horizontal ,vertical or Diagonal directions

TECHNOLOGY USED:- javafx,fxml,java





**LEARNING OUTCOMES:**

I have learnt various concepts in java programming language and I’ve learned main effective Concepts of java such as classes, inheritance ,polymorphism ,collections abstraction and java development.in this I’ve learned how to use JAVAFX to make applications using fxml and java and on final end of the course ive built beautiful game application which is knowns as connectfour

**CONCLUSIONS**

Practical knowledge means the visualization of the knowledge, which we read in our books. For this, we perform experiments and get observations. Practical knowledge is very important in every field. One must be familiar with the problems related to that field so that he may solve them and become a successful person.

After achieving the proper goal in life, an engineer has to enter in professional life. According to this life, he has to serve an industry, may be public or private sector or self-own. For the efficient work in the field, he must be well aware of the practical knowledge as well as theoretical knowledge.

Due to all above reasons and to bridge the gap between theory and practical, our Engineering curriculum provides a practical training of 45 days. During this period a student work in the industry and get well all type of experience and knowledge about the working of companies and hardware and software tools.

I have undergone my 45 days summer training i at **INTERNSHALA** ONLINE TRAININGS This report is based on the knowledge, which I acquired during my 45 days of summer training.

**REFERENCES**

* The javatpoint
* OopS Concept by Sumitra Arora
* Thinking In java (2nd edition)
* Google Group-java TechZone
* **https://docs.oracle.com/en/java/**