ABSTRACT

This report includes the concepts covered during the ESD training programme undertaken at Zensar Technologies. It covers Core Java and PL/SQL. Both of these topics plays a vital role in the software development which is essential for every developer to know. Java is used for programming front end as well as back end and PL/SQL is used to access and manipulate data stored in databases whenever it is required by the developed application or software.

The ESD programme also helped me in learning non-technical knowledge as well. As much as technical knowledge is necessary, non-technical knowledge also plays an important role in shaping an individual for the industry. It included Email writing, Extempore, how to give your introduction, how to dress while working in an organization, etc. All these skills are essential to compete in the industrial world.

ABOUT THE COMPANY

Zensar Technologies is an information technology services and infrastructure services provider with headquarters in Pune, India.

Zensar has operations and a customer base spanning across 29 global locations.
It was founded in 1991 and the current CEO of Zensar Technologies is Sandeep Kishore.It provides various services such as mobility, business process, business application services, testing and assurance, etc. It currently stands with a huge revenue of $460 million and employs 8300+ professionals.

It has subsidiaries in 11 countries including India, UK, Netherlands, Germany, Austria, etc.

Vision

Leaders in business transformation

Mission

We will be the best in delivering innovative industry-focused solutions with measurable business outcomes.

Values

1. Customer Centricity
2. Commitment to People and Community
3. Continuous Innovation and Excellence

Zensar values a lot on how their work environment is and how it is impacting their employees' performance and to improve this they are continuously making it more comfortable and convenient.

Soft Skills Training

Soft skills are a combination of people skills, social skills, communication skills, character traits, attitudes, career attributes, social intelligence and emotional intelligence quotients among others that enable people to effectively navigate their environment, work well with others, perform well, and achieve their goals with complementing hard skills.

The term "soft skills" is defined as "desirable qualities for certain forms of employment that do not depend on acquired knowledge: they include common sense, the ability to deal with people, and a positive flexible attitude."

2.1 Self Introduction

All the students were taught how to properly introduce yourself.

2.2 Eye-Contact Activity

The activity was done in order to build confidence and engage with the listener.

2.3 E – Mail Writing

E – mails are generally shorter than letters. As they are often written quickly, in response to a request or question, they may contain only a few lines.

2.4 Poster Making

We were given a topic and asked to draw anything comes in your mind related to that topic.

2.5 Group Discussion

A group discussion is a group of individuals with similar interest who gather either formally or informally to bring up ideas, solve problems or give comments.

2.6 Paper Tower Model

In this activity, we were given some pages of newspaper and asked to create a structure in vertical direction without using gum. The group whose structure will be greater in height, would win.

2.7 Silent Act

Silent act is acting out a story through body motions, without use of speech. It is the act with no synchronized recorded sound, especially with no dialogue.

PL/SQL Training

3.1 What is PL/SQL?

PL/SQL (Procedural Language/Structured Query Language) is Oracle Corporation's procedural extension for SQL and the Oracle relational database.

PL/SQL is a combination of SQL along with the procedural features of programming languages. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL. PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java.

PL/SQL is a procedural language designed specifically to embrace SQL statements within its syntax. PL/SQL program units are compiled by the Oracle Database server and stored inside the database. And at run-time, both PL/SQL and SQL run within the same server process, bringing optimal efficiency. PL/SQL automatically inherits the robustness, security, and portability of the Oracle Database.

Why do we need PL/SQL when we have SQL?

An application that uses Oracle Database is worthless unless only correct and complete data is persisted. The time-honoured way to ensure this is to expose the database only via an interface that hides the implementation details – the tables and the SQL statements that operate on these. This approach is generally called the thick database paradigm, because PL/SQL subprograms inside the database issue the SQL statements from code that implements the surrounding business logic; and because the data can be changed and viewed only through a PL/SQL interface.

Introduction of PL/SQL

PL/SQL is a completely portable, high-performance transaction-processing language. It provides a built-in, interpreted and OS independent programming environment. It can also directly be called from the command-line SQL*Plus interface.

PL/SQL's general syntax is based on that of ADA and Pascal programming language.

Features of PL/SQL

PL/SQL has the following features −

- PL/SQL is tightly integrated with SQL.
- It offers extensive error checking.
- It offers numerous data types.

- It offers a variety of programming structures.
- It supports structured programming through functions and procedures.
- It supports object-oriented programming.
- It supports the development of web applications and server pages.

Basic Syntax

The Basic Syntax of PL/SQL which is a block-structured language; this means that the PL/SQL programs are divided and written in logical blocks of code. Each block consists of three sub-parts −

Declarations

1 This section starts with the keyword **DECLARE**. It is an optional section and defines all variables, cursors, subprograms, and other elements to be used in the program.

Executable Commands

2 This section is enclosed between the keywords **BEGIN** and **END** and it is a mandatory section.

Exception Handling

3 This section starts with the keyword **EXCEPTION**. This optional section contains **exception(s)** that handle errors in the program.

Following is the basic structure of a PL/SQL block −

```
DECLARE
  <declarations section>
BEGIN
  <executable command(s)>
EXCEPTION
  <exception handling>
END;
```

Every PL/SQL statement ends with a semicolon (;). PL/SQL blocks can be nested within other PL/SQL blocks using BEGIN and END.

Data Types

Scalar

1 Single values with no internal components, such as a **NUMBER, DATE,** or **BOOLEAN**.

Large Object (LOB)

2 Pointers to large objects that are stored separately from other data items, such as text, graphic images, video clips, and sound waveforms.

Composite

3 Data items that have internal components that can be accessed individually. For example, collections and records.

Reference

4 Pointers to other data items.

Most used Data types are: -

1. PLS_INTEGER
2. VARCHAR2
3. FLOAT
4. REAL
5. LONG

Variables

A variable is nothing but a name given to a storage area that our programs can manipulate. Each variable in PL/SQL has a specific data type, which determines the size and the layout of the variable's memory; the range of values that can be stored within that memory and the set of operations that can be applied to the variable.

Variable Declaration in PL/SQL

PL/SQL variables must be declared in the declaration section or in a package as a global variable. When you declare a variable, PL/SQL allocates memory for the variable's value and the storage location is identified by the variable name.

The **syntax** for declaring a variable is −

variable name [CONSTANT] datatype [NOT NULL] [:= | DEFAULT initial_value]

Variable Scope in PL/SQL

PL/SQL allows the nesting of blocks, i.e., each program block may contain another inner block. If a variable is declared within an inner block, it is not accessible to the outer block. However, if a variable is declared and accessible to an outer block, it is also accessible to all nested inner blocks. There are two types of variable scope −

(i) Local variables − Variables declared in an inner block and not accessible to outer blocks.

(ii) Global variables − Variables declared in the outermost block or a package.

Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulation. PL/SQL language is rich in built-in operators and provides the following types of operators −

- Arithmetic operators
- Relational operators
- Comparison operators
- Logical operators
- String operators

Conditional Statements and Loops

PL/SQL programming language provides following types of decision-making statements. Click the following links to check their detail.

1 IF – THEN statement

The IF statement associates a condition with a sequence of statements enclosed by the keywords THEN and END IF. If the condition is true, the statements get executed and if the condition is false or NULL then the IF statement does nothing.

2 IF-THEN-ELSE statement

IF statement adds the keyword ELSE followed by an alternative sequence of statement. If the condition is false or NULL, then only the alternative sequence of statements get executed.

3 Case statement

Like the IF statement, the CASE statement selects one sequence of statements to execute.

PL/SQL provides the following types of loop to handle the looping requirements: -

1. PL/SQL Basic LOOP

In this loop structure, sequence of statements is enclosed between the LOOP and the END LOOP statements. At each iteration, the sequence of statements is executed and then control resumes at the top of the loop.

2. PL/SQL WHILE LOOP

Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

3. PL/SQL FOR LOOP

Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

Constants and Literals

A constant holds a value that once declared, does not change in the program. A constant declaration specifies its name, data type, and value, and allocates storage for it.

A literal is an explicit numeric, character, string, or Boolean value not represented by an identifier.

Procedure

A subprogram is a program unit/module that performs a particular task. These subprograms are combined to form larger programs. A subprogram can be invoked by another subprogram or program which is called the calling program. These subprograms are called Procedures in PL/SQL. These subprograms do not return a value directly; mainly used to perform an action.

A procedure can be created in the following way: -

```
CREATE [OR REPLACE] PROCEDURE procedure_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
{IS | AS}
BEGIN
  < procedure_body >
END procedure_name;
```

Functions

Functions are similar to procedures but they differ on the fact that functions always return some value

```
CREATE [OR REPLACE] FUNCTION function_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
RETURN return_datatype
{IS | AS}
BEGIN
  < function_body >
END [function_name];
```

To use a function, you will have to call that function to perform the defined task. The calling procedure is similar to C or C++ syntax.

Date & Time

DATE

It stores date and time information in both character and number data types. It is made of information on century, year, month, date, hour, minute, and second.

TIMESTAMP

It is an extension of the DATE data type. It stores the year, month, and day of the DATE datatype, along with hour, minute, and second values. It is useful for storing precise time values.

Exceptions

An exception is an error condition during a program execution. PL/SQL supports programmers to catch such conditions using EXCEPTION block in the program and an appropriate action is taken against the error condition. There are two types of exceptions −

(i) System-defined exceptions

(ii) User-defined exceptions

```
DECLARE
  <declarations section>
BEGIN
  <executable command(s)>
EXCEPTION
  <exception handling goes here >
```

```
WHEN exception1 THEN
    exception1-handling-statements
  WHEN others THEN
    exception3-handling-statements
END;
```

Exceptions are raised by the database server automatically whenever there is any internal database error, but exceptions can be raised explicitly by the programmer by using the command RAISE.

Cursors

A cursor is a pointer to context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the active set.

There are two types of cursors −

(i) Implicit cursors

(ii) Explicit cursors

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement. Programmers cannot control the implicit cursors and the information in it.

Explicit cursors are programmer-defined cursors for gaining more control over the context area. An explicit cursor should be defined in the declaration section of the PL/SQL Block. It is created on a SELECT Statement which returns more than one row.

The **syntax** for creating an explicit cursor is −

CURSOR cursor_name IS select_statement;

Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events −

(i) A database manipulation (DML) statement (DELETE, INSERT, or UPDATE)

(ii) A database definition (DDL) statement (CREATE, ALTER, or DROP).

(iii) A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Triggers can be defined on the table, view, schema, or database with which the event is associated.

Benefits of Triggers

1. Generating some derived column values automatically
2. Enforcing referential integrity
3. Event logging and storing information on table access
4. Synchronous replication of tables
5. Imposing security authorizations
6. Preventing invalid transactions

The **syntax** for creating a trigger is −

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
  Declaration-statements
BEGIN
  Executable-statements
EXCEPTION
  Exception-handling-statements
END;
```

The created trigger will be fired whenever we perform the triggering operation in PL/SQL

Java Training

Introduction

Java is a programming language and a platform.

Java is a high level, robust, secured and object-oriented programming language.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has its own runtime environment (JRE) and API, it is called platform.

A simple example in Java to write a Hello World program

```
class Simple {
      public static void main (String args[]){
       System.out.println("Hello Java");
      }
  }
```

According to Sun, 3 billion devices run Java. There are many devices where Java is currently used. Some of them are as follows:

1. Desktop Applications such as acrobat reader, media player, antivirus etc.
2. Web Applications such as irctc.co.in, javatpoint.com etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System

History

Currently, Java is used in internet programming, mobile devices, games, e-business solutions etc.

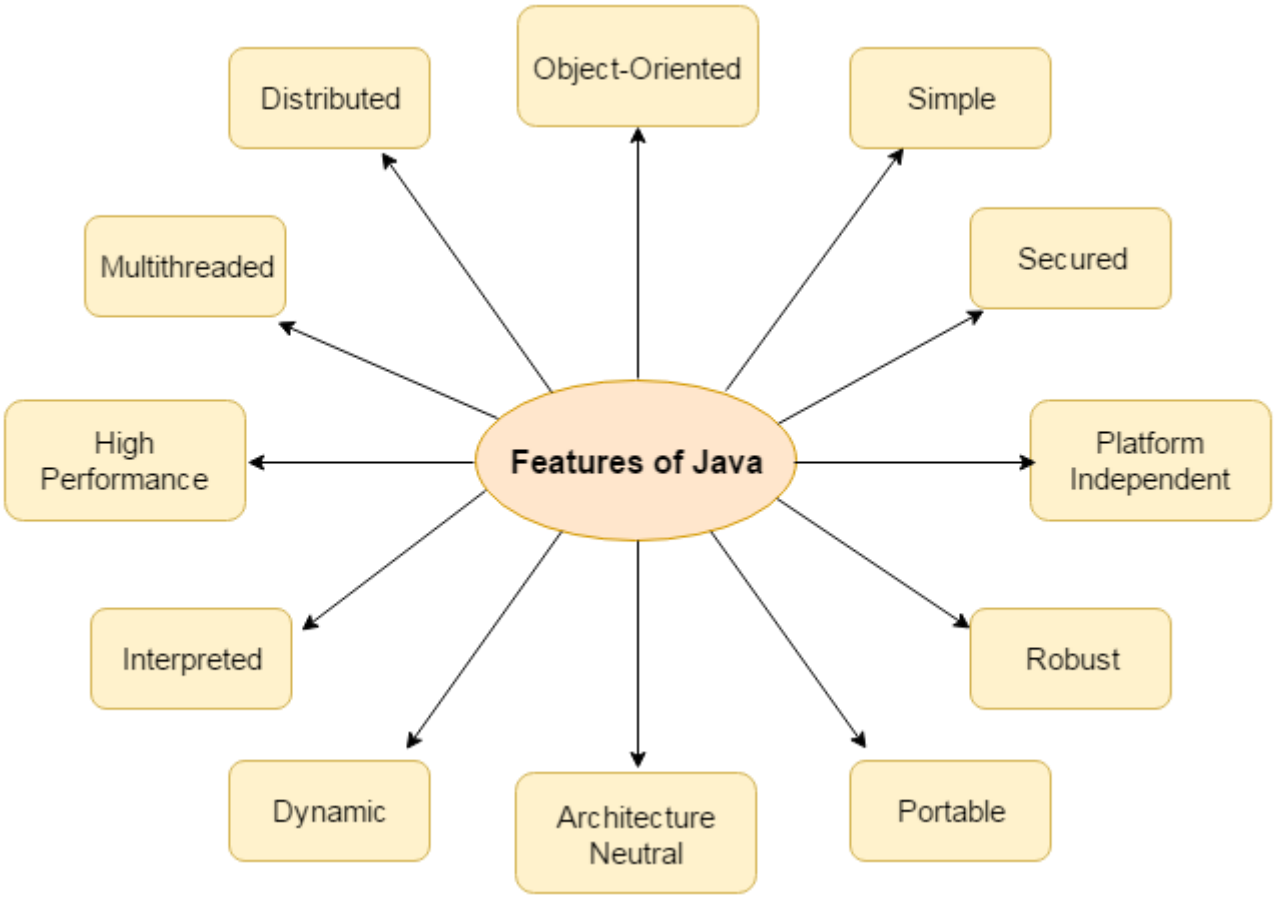
1) **James Gosling**, **Mike Sheridan**, and **Patrick Naughton** initiated the Java language project in June 1991. The small team of sun engineers called **Green Team**.

2) Originally designed for small, embedded systems in electronic appliances like set-top boxes.

3) Firstly, it was called **"Greentalk"** by James Gosling and file extension was .gt.

4) After that, it was called **Oak** and was developed as a part of the Green project.

Features of Java



JVM

JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which Java bytecode can be executed.

JVMs are available for many hardware and software platforms. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent.

The JVM performs following main tasks:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

Variable

Variable is name of *reserved area allocated in memory*. In other words, it is a *name of memory location*. It is a combination of "vary + able" that means its value can be changed.

int data=50; //Here data is variable

Types of Variable

There are three types of variables in Java:

- local variable
- instance variable
- static variable

Data Types in Java with their default values and default size

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

Conditional Statements

The Java *if statement* is used to test the condition. It checks boolean condition: *true* or *false*. There are various types of if statement in Java.

- if statement
- if-else statement

Java IF Statement

The Java if statement tests the condition. It executes the *if block* if condition is true

Syntax:

```
if(condition) {
//code to be executed}
```

Java IF-else Statement

The Java if-else statement also tests the condition. It executes the *if block* if condition is true otherwise *else block* is executed.

Syntax:

```
if(condition) {
//code if condition is true
} else {//code if condition is false}
```

Switch Statement

The Java switch statement executes one statement from multiple conditions. It is like if-else-if ladder statement.

Loops

For Loop

The Java for loop is used to iterate a part of the program several times. If the number of iteration is fixed, it is recommended to use for loop.

There are three types of for loop in Java.

1. Simple For Loop

2. For-each or Enhanced For Loop

3. Labelled For Loop

Simple For Loop

The simple for loop is same as C/C++. We can initialize variable, check condition and increment/decrement value.

Syntax:

for (initialization; condition; incr/decr) { //code to be executed }

For-each Loop

The for-each loop is used to traverse array or collection in java. It is easier to use than simple for loop because we don't need to increment value and use subscript notation. It works on elements basis not index. It returns element one by one in the defined variable.

Syntax:

for (Type var: array) {

//code to be executed}

Labeled For Loop

We can have name of each for loop. To do so, we use label before the for loop. It is useful if we have nested for loop so that we can break/continue specific for loop.

Syntax:

labelname:

for(initialization; condition; incr/decr){//code to be executed}

While Loop

The Java while loop is used to iterate a part of the program several times. If the number of iteration is not fixed, it is recommended to use while loop.

Syntax:

while(condition){

//code to be executed

}

do-while Loop

The Java do-while loop is used to iterate a part of the program several times. If the number of iteration

is not fixed and you must have to execute the loop at least once, it is recommended to use do-while loop.

Syntax:

do{

//code to be executed}while(condition);

OOPS CONCEPTS

Object means a real word entity such as pen, chair, table etc. Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts:

Object

Class

Inheritance

Polymorphism

Abstraction

Encapsulation

Object

Any entity that has state and behaviour is known as an object. For example: chair, pen, table, keyboard, bike etc. It can be physical and logical.

Class

Collection of objects is called class. It is a logical entity.

Inheritance

When one object acquires all the properties and behaviours of parent object i.e. known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

Polymorphism

When one task is performed by different ways i.e. known as polymorphism. For example: to convince the customer differently, to draw something e.g. shape or rectangle etc.

Abstraction

Hiding internal details and showing functionality is known as abstraction. For example: phone call, we don't know the internal processing.

Encapsulation

Binding (or wrapping) code and data together into a single unit is known as encapsulation

Naming conventions

Java naming convention is a rule to follow as you decide what to name your identifiers such as class, package, variable, constant, method etc.

But, it is not forced to follow. So, it is known as convention not rule.

Name	Convention
class name	should start with uppercase letter and be a noun e.g. String, Color, Button, System, Thread etc.
interface name	should start with uppercase letter and be an adjective e.g. Runnable, Remote, ActionListener etc.
method name	should start with lowercase letter and be a verb e.g. actionPerformed(), main(), print(), println() etc.
variable name	should start with lowercase letter e.g. firstName, orderNumber etc.
package name	should be in lowercase letter e.g. java, lang, sql, util etc.
constants name	should be in uppercase letter. e.g. RED, YELLOW, MAX_PRIORITY etc.

CamelCase in java naming conventions

Java follows camelcase syntax for naming the class, interface, method and variable.

If name is combined with two words, second word will start with uppercase letter always e.g. actionPerformed(), firstName, ActionEvent, ActionListener etc.

Constructor in Java

Constructor in java is a special type of method that is used to initialize the object.

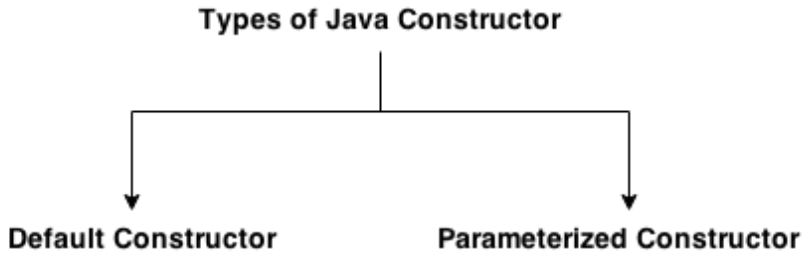
Java constructor is invoked at the time of object creation. It constructs the values i.e. provides data for the object that is why it is known as constructor.

Rules for creating java constructor

There are basically two rules defined for the constructor.

1. Constructor name must be same as its class name

2. Constructor must have no explicit return type



Java Default Constructor

A constructor that have no parameter is known as default constructor.

Syntax of default constructor:

<class_name>(){}

Java parameterized constructor

A constructor that have parameters is known as parameterized constructor.

Why use parameterized constructor?

Parameterized constructor is used to provide different values to the distinct objects.

this keyword in java

There can be a lot of usage of java this keyword. In java, this is a reference variable that refers to the current object.

Usage of java this keyword

Here is given the 6 usage of java this keyword.

1. this can be used to refer current class instance variable.
2. this can be used to invoke current class method (implicitly)
3. this () can be used to invoke current class constructor.
4. this can be passed as an argument in the method call.
5. this can be passed as argument in the constructor call.
6. this can be used to return the current class instance from the method.

Inheritance in Java

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviours of parent object.

The idea behind inheritance in java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of parent class, and you can add new methods and fields also.

Why use inheritance in Java?

1. For Method Overriding (so runtime polymorphism can be achieved).
2. For Code Reusability.

Syntax of Java Inheritance

```
class Subclass-name extends Superclass-name
{   //methods and fields }
```

Types of inheritance in Java

1. Single
2. Multilevel
3. Hierarchical
4. Multiple
5. Hybrid

Method Overloading

If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.

Advantage of method overloading

1. Method overloading increases the readability of the program.
2. Different ways to overload the method

There are two ways to overload the method in java

1. By changing number of arguments
2. By changing the data type

Method Overriding

If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in Java

Usage of Java Method Overriding

1. Method overriding is used to provide specific implementation of a method that is already provided by its super class.
2. Method overriding is used for runtime polymorphism

Rules for Java Method Overriding

1. Method must have same name as in the parent class
2. Method must have same parameter as in the parent class.
3. Must be IS-A relationship (inheritance).

super keyword

The super keyword in java is a reference variable which is used to refer immediate parent class object. Whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.

Usage of Java super Keyword

1. super can be used to refer immediate parent class instance variable.
2. super can be used to invoke immediate parent class method.
3. super () can be used to invoke immediate parent class constructor.

Abstract class

A class that is declared with abstract keyword, is known as abstract class in java. It can have abstract and non-abstract methods (method with body).

Abstraction

Abstraction is a process of hiding the implementation details and showing only functionality to the user.

Another way, it shows only important things to the user and hides the internal details for example sending SMS, you just type the text and send the message. You don't know the internal processing about the message delivery.

Abstraction lets you focus on what the object does instead of how it does it.

Ways to achieve Abstraction

There are two ways to achieve abstraction in Java

1. Abstract class (0 to 100%)
2. Interface (100%)

Abstract class

A class that is declared as abstract is known as abstract class. It needs to be extended and its method implemented. It cannot be instantiated.

Example abstract class

```
abstract class A{}
```

Abstract method

A method that is declared as abstract and does not have implementation is known as abstract method.

Example abstract method

```
abstract void printStatus();//no body and abstract
```

Interface

An interface in Java is a blueprint of a class. It has static constants and abstract methods.

The interface in java is a mechanism to achieve abstraction. There can be only abstract methods in the java interface not method body. It is used to achieve abstraction and multiple inheritance in Java.

Java Interface also represents IS-A relationship.

It cannot be instantiated just like abstract class

Why use Java interface?

There are mainly three reasons to use interface. They are given below.

1. It is used to achieve abstraction.
2. By interface, we can support the functionality of multiple inheritance.
3. It can be used to achieve loose coupling.

Java Package

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

Package in java can be categorized in two form, built-in package and user-defined package.

There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Here, we will have the detailed learning of creating and using user-defined packages.

Advantage of Java Package

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.

2) Java package provides access protection.

3) Java package removes naming collision.

Access Modifiers

There are two types of modifiers in java: access modifiers and non-access modifiers.

The access modifiers in java specifies accessibility (scope) of a data member, method, constructor or class.

There are 4 types of java access modifiers:

1. private
2. default
3. protected
4. public

There are many non-access modifiers such as static, synchronized, native, volatile, transient etc.

1) private access modifier

The private access modifier is accessible only within class.

2) default access modifier

If you don't use any modifier, it is treated as default by default. The default modifier is accessible only within package.

3) protected access modifier

The protected access modifier is accessible within package and outside the package but through inheritance only.

4) public access modifier

The public access modifier is accessible everywhere. It has the widest scope among all other modifiers.

Encapsulation

Encapsulation in java is a process of wrapping code and data together into a single unit, for example capsule i.e. mixed of several medicines.

We can create a fully encapsulated class in java by making all the data members of the class private. Now we can use setter and getter methods to set and get the data in it.

The Java Bean class is the example of fully encapsulated class.

Advantage of Encapsulation in java

By providing only setter or getter method, you can make the class read-only or write-only.

It provides you the control over the data. Suppose you want to set the value of id i.e. greater than 100 only, you can write the logic inside the setter method.

Wrapper classes

Wrapper class in java provides the mechanism to convert primitive into object and object into primitive.

Since J2SE 5.0, autoboxing and unboxing feature converts primitive into object and object into primitive automatically. The automatic conversion of primitive into object is known as autoboxing and vice-versa unboxing.

The eight classes of java.lang package are known as wrapper classes in java. The list of eight wrapper classes are given below:

Primitive Type	Wrapper class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Immutable Strings

In java, string objects are immutable. Immutable simply means unmodifiable or unchangeable. Once string object is created its data or state can't be changed but a new string object is created.

StringBuffer class

StringBuffer class is used to create mutable (modifiable) string. The StringBuffer class in java is same as String class except it is mutable i.e. it can be changed.

Methods in StringBuffer class :-

1. append()
2. insert()
3. replace()
4. delete()
5. reverse()
6. capacity()

StringBuilder class

StringBuilder class is used to create mutable (modifiable) string. The Java StringBuilder class is same as StringBuffer class except that it is non-synchronized. It is available since JDK 1.5

toString() method

If you want to represent any object as a string, toString() method comes into existence.

The toString() method returns the string representation of the object.

If you print any object, java compiler internally invokes the toString() method on the object. So overriding the toString() method, returns the desired output, it can be the state of an object etc. depends on your implementation.

Advantage of Java toString() method

By overriding the toString() method of the Object class, we can return values of the object, so we don't need to write much code.

Various String class methods are:-

1. charAt()
2. compareTo()
3. concat()
4. contains()
5. endsWith()
6. equals()
7. format()
8. getBytes()
9. length()
10. split()

Exception Handling

The exception handling in java is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained.

In java, exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

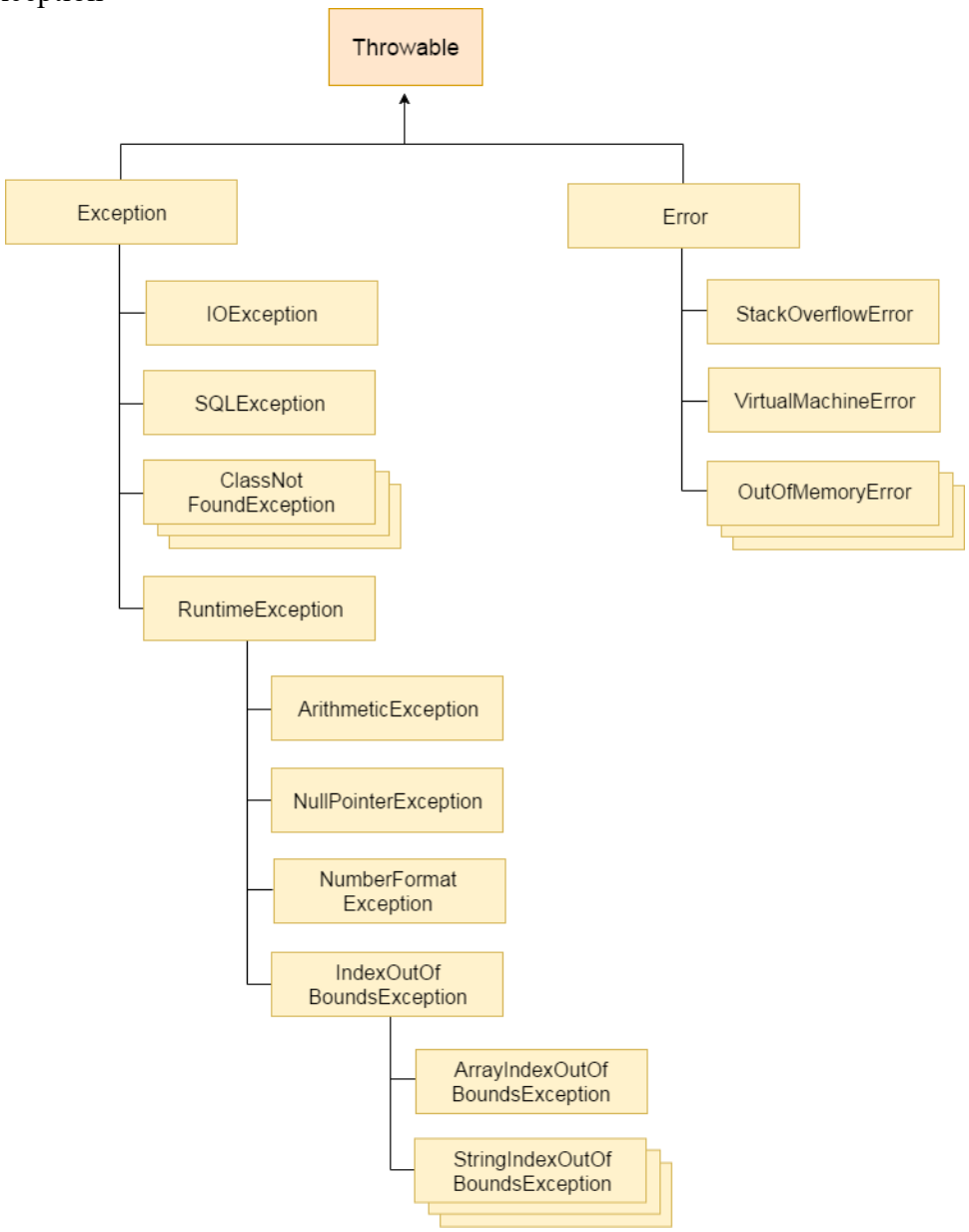
What is exception handling?

Exception Handling is a mechanism to handle runtime errors such as ClassNotFound, IO, SQL, Remote etc.

Advantage of Exception Handling

The core advantage of exception handling is to maintain the normal flow of the application. Exception normally disrupts the normal flow of the application that is why we use exception handling

Types of Exception



There are 5 keywords used in java exception handling.

1. try
2. catch
3. finally
4. throw
5. throws

Syntax of java try-catch

try{

//code that may throw exception

}catch(Exception_class_Name ref){}

Multithreading

Multithreading in java is a process of executing multiple threads simultaneously.

Thread is basically a lightweight sub-process, a smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.

But we use multithreading than multiprocessing because threads share a common memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.

Java Multithreading is mostly used in games, animation etc.

Advantages of Java Multithreading

1) It doesn't block the user because threads are independent and you can perform multiple operations at same time.

2) You can perform many operations together so it saves time.

3) Threads are independent so it doesn't affect other threads if exception occur in a single thread.

Frequently used method for threads are:-

1. run()
2. sleep()
3. start()
4. setPriority()

I/O

Java I/O (Input and Output) is used to process the input and produce the output.

Java uses the concept of stream to make I/O operation fast. The java.io package contains all the classes required for input and output operations.

We can perform file handling in java by Java I/O API.

Stream

A stream is a sequence of data. In Java a stream is composed of bytes. It's called a stream because it is like a stream of water that continues to flow.

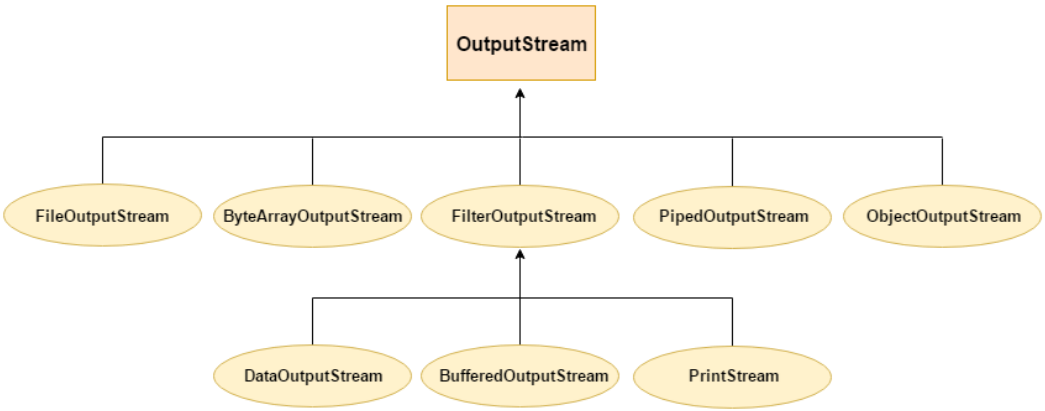
1) System.out: standard output stream

2) System.in: standard input stream

3) System.err: standard error stream

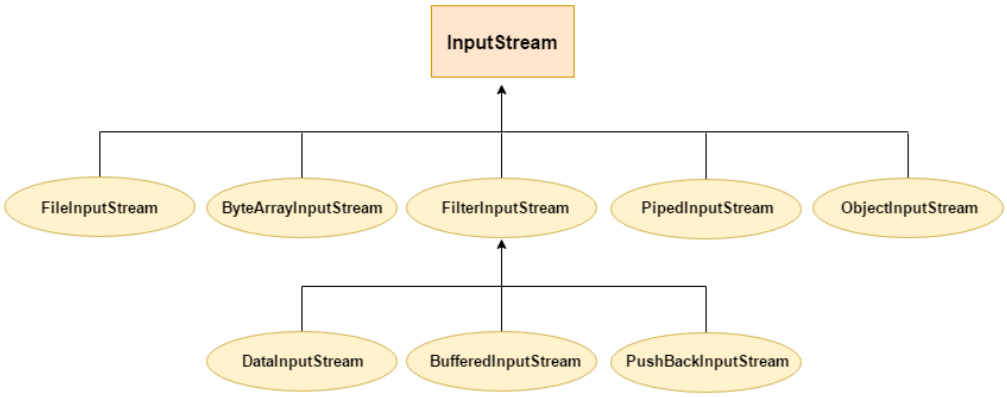
OutputStream

Java application uses an output stream to write data to a destination, it may be a file, an array, peripheral device or socket.



InputStream

Java application uses an input stream to read data from a source, it may be a file, an array,peripheral device or socket



Serialization

Serialization in java is a mechanism of writing the state of an object into a byte stream.

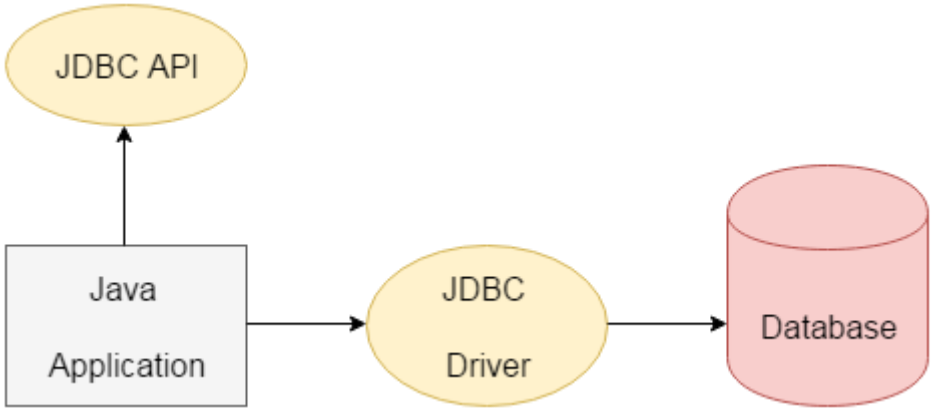
java.io.Serializable interface

Serializable is a marker interface (has no data member and method). It is used to "mark" java classes so that objects of these classes may get certain capability. The Cloneable and Remote are also marker interfaces.

It must be implemented by the class whose object you want to persist.

Java JDBC

Java JDBC is a java API to connect and execute query with the database. JDBC API uses jdbc drivers to connect with the database.



Why use JDBC?

Before JDBC, ODBC API was the database API to connect and execute query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

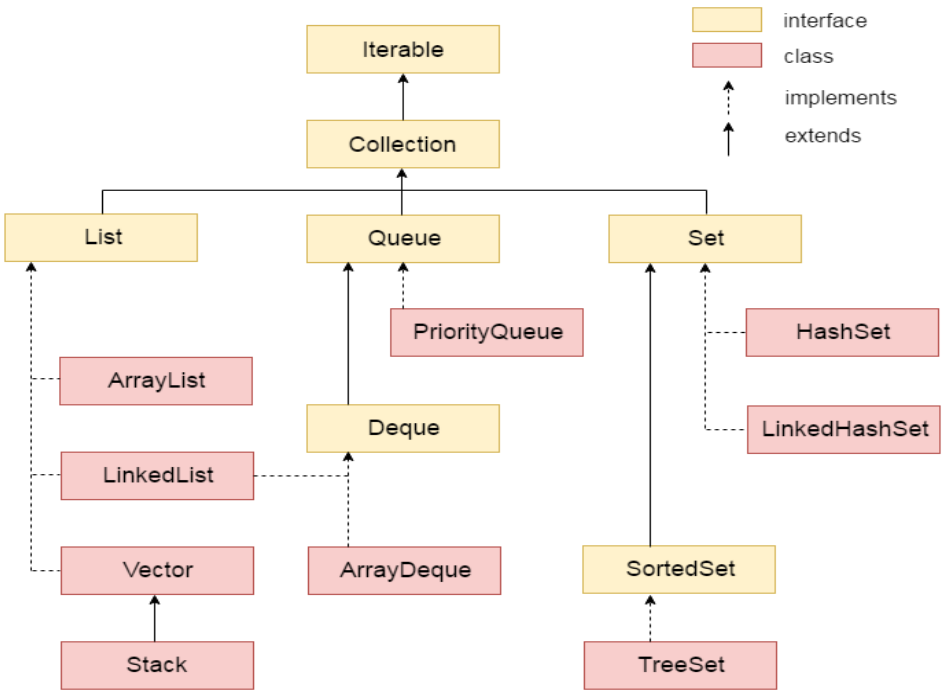
Collections

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).

Collection represents a single unit of objects i.e. a group.



There are many methods declared in the Collection interface. They are as follows:

No.	Method	Description
1	public boolean add(Object element)	is used to insert an element in this collection.
2	public boolean addAll(Collection c)	is used to insert the specified collection elements in the invoking collection.
3	public boolean remove(Object element)	is used to delete an element from this collection.
4	public boolean removeAll(Collection c)	is used to delete all the elements of specified collection from the invoking collection.
5	public boolean retainAll(Collection c)	is used to delete all the elements of invoking collection except the specified collection.
6	public int size()	return the total number of elements in the collection.
7	public void clear()	removes the total no of element from the collection.

Bibliography

1. JavaTpoint – www.javatpoint.com

2. TutorialsPoint.com – www.tutorialspoint.com

3. Oracle DB 12C – www.oracle.com

4. Java a primer by E Balaguruswamy