

MOVIE RECOMMENDATION SYSTEM

Project Submitted in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Technology in the field of Computer Science and Engineering

BY
ABHISHEK SINGH (123200903003)
SAKSHEE (123200903106)
AGNIBHA MAITY(123200903011)

Under the supervision
of
MR. SANKET DAN



Department of Computer Science and Engineering
JIS College of Engineering

Block-A, Phase-III, Kalyani, Nadia, Pin-741235
West Bengal, India
May, 2024



JIS College of Engineering

Block 'A', Phase-III, Kalyani, Nadia, 741235
Phone: +91 33 2582 2137, Telefax: +91 33 2582 2138
Website: www.jiscollege.ac.in, Email: info@jiscollege.ac.in

CERTIFICATE

This is to certify that **SAKSHEE(123200903106), ABHISHEK SINGH (123200903003), AGIBHA MAITY (123200903011)**. has completed their project entitled **Movie Recommendation System**, under the guidance of **Mr. Sanket Dan** in partial fulfilment of the requirements for the award of the **Bachelor of Technology in Computer Science and Engineering** from JIS college of Engineering (An Autonomous Institute) is an authentic record of their own work carried out during the academic year 2023-2024 and to the best of our knowledge, this work has not been submitted elsewhere as part of the process of obtaining a degree, diploma, fellowship or any other similar title.

Signature of the Supervisor Signature of the HOD Signature of the Principal

Place: KALYANI

Date: 21/04/2024

ACKNOWLEDGEMENT

The analysis of the project work wishes to express our gratitude to Mr. Sanket Dan for allowing the degree attitude and providing effective guidance in development of this project work. His conscription of the topic and all the helpful hints, he provided, contributed greatly to successful development of this work, without being pedagogic and overbearing influence. We also express our sincere gratitude to Dr. Bikramjit Sarkar, Head of the Department of Computer Science and Engineering of JIS College of Engineering and all the respected faculty members of Department of CSE for giving the scope of successfully carrying out the project work. Finally, we take this opportunity to thank to Prof. **(Dr.) Partha Sarkar**, Principal of JIS College of Engineering for giving us the scope of carrying out the project work.

Date:

Team members:

ABHISHEK SINGH

B.TECH in Computer Science And Engineering

4th Year /8th SEMESTER

Univ Roll-123200903003

SAKSHEE

B.TECH in Computer Science And Engineering

4th Year /8th SEMESTER

Univ Roll-123200903106

AGNIBHA MAITY

B.TECH in Computer Science And Engineering

4th Year /8th SEMESTER

Univ Roll-123200903011

ABSTRACT

In this hustling world, entertainment is a necessity for each one of us to refresh our mood and energy. Entertainment regains our confidence for work and we can work more enthusiastically. For revitalizing ourselves, we can listen to our preferred music or can watch movies of our choice. For watching favourable movies online we can utilize movie recommendation systems, which are more reliable, since searching of preferred movies will require more and more time which one cannot afford to waste. In this paper, to improve the quality of a movie recommendation system, a Hybrid approach by combining content based filtering and collaborative filtering, using Support Vector Machine as a classifier and genetic algorithm is presented in the proposed methodology and comparative results have been shown which depicts that the proposed approach shows an improvement in the accuracy, quality and scalability of the movie recommendation system than the pure approaches in three different datasets. Hybrid approach helps to get the advantages from both the approaches as well as tries to eliminate the drawbacks of both methods.

TABLE OF CONTENTS

	Page No.
Certificate	2
Declaration	3Error! Bookmark not defined.
Abstract	4
Acknowledgement	5
Table of contents	5
List of Figures	7
List of Tables	9
List of Abbreviations	x
1 INTRODUCTION	1
1.1 Relevance of the Project	1
1.2 Problem Statement	2
1.3 Objective	2
1.4 Scope of the Project	3
1.5 Methodology	
2 LITERATURE SURVEY	4
2.1 k-means and k-nearest	4
2.2 Using Collaborative	5
3 SYSTEM REQUIREMENTS SPECIFICATION	6
3.1 Hardware Requirements	6
3.2 Software Specification	6
3.3 Software Requirements	6
3.3.1 Anaconda distribution	6
3.3.2 Python Libraries	7

4 SYSTEM ANALYSIS AND DESIGN	8
4.1 System Architecture	17
4.2 Activity diagram	18
4.3 Flowchart	17
5 IMPLEMENTATION	11
5.1 Cosine similarity	11
5.2 Singular Value Decomposition	11
5.3 Experimental Setup	12
6 RESULTS AND DISCUSSION	14
6.1 Screenshots	15
7 TESTING	16
7.1 Testing Methodologies	17
8 CONCLUSION AND FUTURE SCOPE	26
8.1 Conclusion	26
8.2 Future Scope	26
REFERENCES	28

LIST OF FIGURES

	Page No.
Fig 4.1 Architecture for hybrid approach	8
Fig 4.2 Activity diagram	18
Fig 4.3 Data Flow	19
Fig 5.1 Code snippet	22
Fig 5.2 Backed code snippet	23
Fig 6.2 List of recommended movies	23

LIST OF TABLES

	Page No.
TABLE 1: Approach for Different System	24
TABLE 2: RMSE with Proposed Technique	24
TABLE 3: RMSE with Existing Technique	24

CHAPTER 1

INTRODUCTION

1.1 Relevance of the Project

A recommendation system or recommendation engine is a model used for information filtering where it tries to predict the preferences of a user and provide suggestions based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places and other utilities. These systems collect information about a user's preferences and behaviour, and then use this information to improve their suggestions in the future.

Movies are a part and parcel of life. There are different types of movies like some for entertainment, some for educational purposes, some are animated movies for children, and some are horror movies or action films. Movies can be easily differentiated through their genres like comedy, thriller, animation, action etc. Other way to distinguish among movies can be either by releasing year, language, director etc. Watching movies online, there are a number of movies to search in our most liked movies . Movie Recommendation Systems helps us to search our preferred movies among all of these different types of movies and hence reduce the trouble of spending a lot of time searching our favourable movies. So, it requires that the movie recommendation system should be very reliable and should provide us with the recommendation of movies which are exactly same or most matched with our preferences.

A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffers with poor recommendation quality and scalability issues.

1.2 Problem Statement:

The goal of the project is to recommend a movie to the user.

Providing related content out of relevant and irrelevant collection of items to users of online service providers.

1.3 Objective of the Projects

- Improving the Accuracy of the recommendation system
- Improve the Quality of the movie Recommendation system .
- Improving the Scalability.
- Enhancing the user experience.

1.4 Scope of the Project

The objective of this project is to provide accurate movie recommendations to users. The goal of the project is to improve the quality of movie recommendation system, such as accuracy, quality and scalability of system than the pure approaches. This is done using Hybrid approach by combining content based filtering and collaborative filtering, To eradicate the overload of the data, recommendation system is used as information filtering tool in social networking sites .Hence, there is a huge scope of exploration in this field for improving scalability, accuracy and quality of movie recommendation systems Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffers with poor recommendation quality and scalability issues.

1.5 Methodology for Movie Recommendation

The hybrid approach proposed an integrative method by merging fuzzy K-means clustering method and genetic algorithm based weighted similarity measure to construct a movie recommendation system. The proposed movie recommendation system gives finer similarity metrics and quality than the

existing Movie recommendation system but the computation time which is taken by the proposed recommendation system is more than the existing recommendation system. This problem can be fixed by taking the clustered data points as an input dataset

The proposed approach is for improving the scalability and quality of the movie recommendation system .We use a Hybrid approach , by unifying Content-Based Filtering and Collaborative Filtering, so that the approaches can be profited from each other. For computing similarity between the different movies in the given dataset efficiently and in least time and to reduce computation time of the movie recommender engine we used cosine similarity measure.

Agile Methodology:

1.Collecting the data sets: Collecting all the required data set from Kaggle web site.in this project we require movie.csv,ratings.csv,users.csv.

2.Data Analysis: make sure that that the collected data sets are correct and analysing the data in the csv files. i.e. checking whether all the column Fields are present in the data sets.

3.Algorithms: in our project we have only two algorithms one is cosine similarity and other is single valued decomposition are used to build the machine learning recommendation model.

4.Training and Testing the Model: *once the implementation of* algorithm is completed . we have to train the model to get the result. We have tested it several times the model is recommend different set of movies to different users.

5.Improvements in the project: In the later stage we can implement different algorithms and methods for better recommendation.

CHAPTER 2

LITERATURE SURVEY

Over the years, many recommendation systems have been developed using either collaborative, content based or hybrid filtering methods. These systems have been implemented using various big data and machine learning algorithms. Content based [1], [2] collaborative [3] and hybrid [4] are the different approaches used by past researcher for the development of recommender system. In 2007 a web-based movie recommendation system using hybrid filtering methods is presented by the authors [5]. In 2011 a movie recommendation system based on genre correlations is proposed by the authors [6].In 2013 a Bayesian network and Trust model based movie recommendation system is proposed, the Bayesian network is imported for user preference modeling and trust model is used to filter the recommending history data and enable the system to tolerant the noisy data [7].In 2016, authors proposed Recommender systems to predict the rating for users and items, predominantly from big data to recommend their likes. Movie recommendation systems provide a mechanism to assist users in classifying users with similar interests. This system (K-mean Cuckoo) has 0.68 MAE [8], [9].In 2017 authors used a new approach that can solve sparsity problem to a great extent[10].In 2018, authors built a recommendation engine by analyzing rating data sets collected from Twitter to recommend movies to specific user using R[39].

2.1 Movie Recommendation System by K-Means Clustering AND K-Nearest Neighbour

A recommendation system collect data about the user's preferences either implicitly or explicitly on different items like movies. An implicit acquisition in the development of movie recommendation system uses the user's behaviour while watching the movies. On the other hand, a explicit acquisition in the development of movie recommendation system uses the user's previous ratings or history. The other supporting technique that are used in the development of recommendation system is clustering. Clustering is a process to group a set of objects in such a way that objects in the same clusters are more similar to each other than to those in other clusters. K-Means Clustering along with K-Nearest Neighbour is implemented on the movie lens dataset in order to obtain the best-optimized result. In existing technique, the data is

scattered which results in a high number of clusters while in the proposed technique data is gathered and results in a low number of clusters. The process of recommendation of a movie is optimized in the proposed scheme. The proposed recommender system predicts the user's preference of a movie on the basis of different parameters. The recommender system works on the concept that people are having common preference or choice. These users will influence on each other's opinions. This process optimizes the process and having lower RMSE.

2.2 Movie Recommendation System Using Collaborative

Filtering: By Ching-Seh (Mike) Wu,Deepti Garg,Unnathi Bhandary

Collaborative filtering systems analyse the user's behaviour and preferences and predict what they would like based on similarity with other users. There are two kinds of collaborative filtering systems; user-based recommender and item-based recommender.

1.Use-based filtering: User-based preferences are very common in the field of designing personalized systems. This approach is based on the user's likings. The process starts with users giving ratings (1-5) to some movies. These ratings can be implicit or explicit. Explicit ratings are when the user explicitly rates the item on some scale or indicates a thumbs-up/thumbs-down to the item. Often explicit ratings are hard to gather as not every user is much interested in providing feedbacks. In these scenarios, we gather implicit ratings based on their behaviour. For instance, if a user buys a product more than once, it indicates a positive preference. In context to movie systems, we can imply that if a user watches the entire movie, he/she has some likeability to it. Note that there are no clear rules in determining implicit ratings. Next, for each user, we first find some defined number of nearest neighbours. We calculate correlation between users' ratings using Pearson Correlation algorithm. The assumption that if two users' ratings are highly correlated, then these two users must enjoy similar items and products is used to recommend items to users.

2.Item-based filtering: Unlike the user-based filtering method, item based focuses on the similarity between the item's users like instead of the users themselves. The most similar items are computed ahead of time. Then for recommendation, the items that are most similar to the target item are recommended to the user.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

This chapter involves both the hardware and software requirements needed for the project and detailed explanation of the specifications.

3.1 Hardware Requirements

- A PC with Windows/Linux OS
- Processor with 1.7-2.4GHz speed
- Minimum of 8gb RAM
- 2gb Graphic card

3.2 Software Specification

- Text Editor (VS-code/WebStorm)
- Anaconda distribution package (PyCharm Editor)
- Python libraries

3.3 Software Requirements

3.3.1 **Anaconda distribution:**

Anaconda is a free and open-source distribution of the Python programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management system and deployment. Package versions are managed by the package management system conda. The anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.3

3.3.3 **Python libraries:**

For the computation and analysis we need certain python libraries which are used to perform analytics. Packages such as SkLearn, Numpy, pandas, Matplotlib, Flask framework, etc are needed.

Sklearn: It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. **Pandas:** Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.

Flask: It is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 System Architecture of Proposed System:

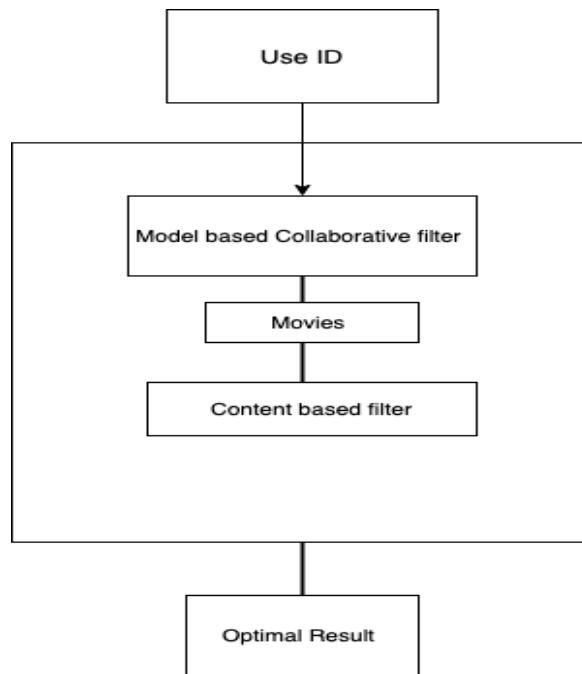


Fig:-4.1 Architecture for hybrid approach

For each different individual user different list of movies are recommended ,as user login or enters the user id based on two different approaches used in the project each will recommend the set of movies to the particular user by combining the both the set of movie based on the user the hybrid model will recommend the single list of movie to the user.

Activity Diagram:

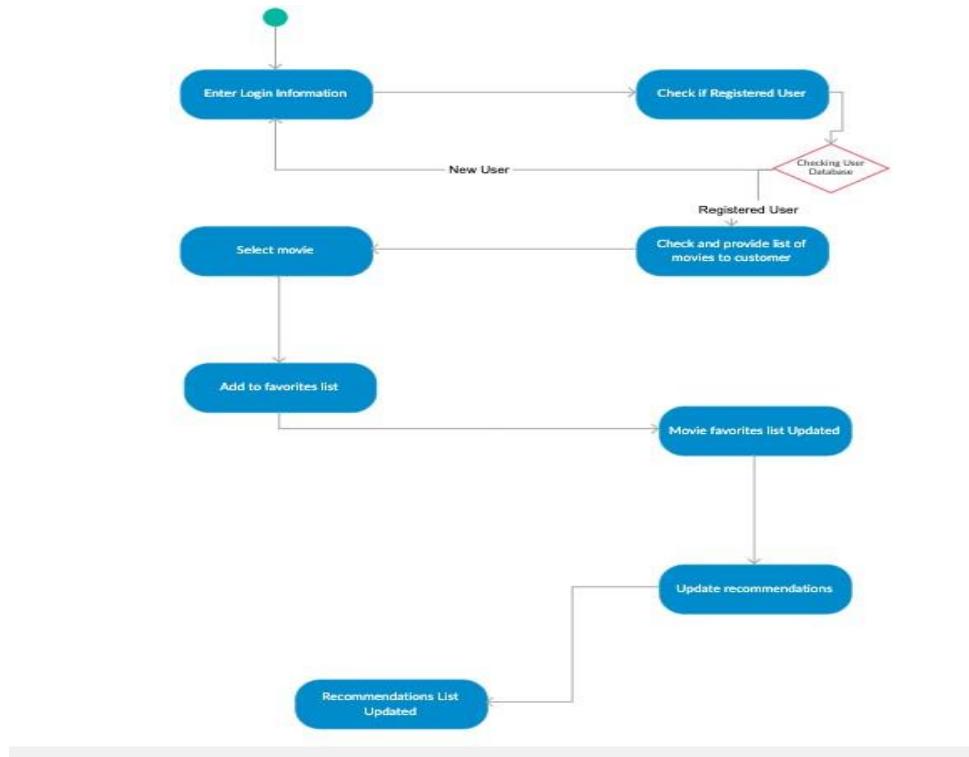


Fig:-4.2 Activity diagram

Once the user login by entering the user-id i.e present in the csv file ranges from 15000 the list of movie are recommended to the user .

4.3 Dataflow:

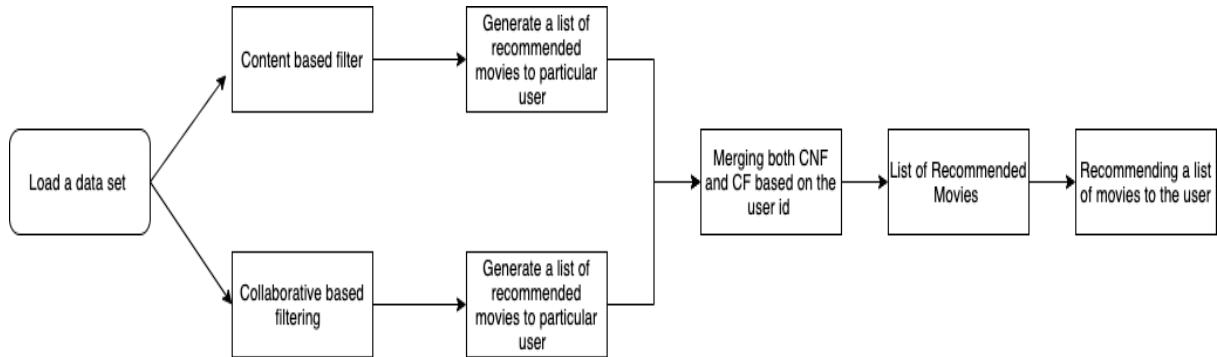


Fig:-4.3 Data Flow Diagram

Initially load the data sets that are required to build a model the data set that are required in this project are movies.csv, ratinfg.csv, users.csv all the data sets are available in the Kaggle.com. Basically, two models are built in this project content based and collaborative filtering each produce a list of movies to a particular user by combining both based on the user id a single final list of movies are recommended to the particular user .

CHAPTER 5

IMPLEMENTATION

The Proposed System Make Use Different Algorithms and Methods for the implementation of Hybrid Approach

5.1 Cosine Similarity: **Cosine similarity** is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.

Formula:

$$\text{Cos}\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$ is the dot product of the two vectors.

5.2 Singular Value Decomposition (SVD):

Let A be an $n \times d$ matrix with singular vectors v_1, v_2, \dots, v_r and corresponding singular values $\sigma_1, \sigma_2, \dots, \sigma_r$. Then $u_i = (\sigma_i^{-1}) A v_i$, for $i = 1, 2, \dots, r$, are the left singular vectors and by Theorem 1.5, A can be decomposed into a sum of rank one matrices a

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T.$$

We first prove a simple lemma stating that two matrices A and B are identical if $Av = Bv$ for all v. The lemma states that in the abstract, a matrix A can be viewed as a transformation that maps vector v onto Av

CHAPTER 6

RESULTS AND DISCUSSION

Since our project is movie recommendation system .one can develop a movie recommendation system by using either content based or collaborative filtering or combining both.

In our project we have developed a hybrid approach i.e combination of both content and collaborative filtering .Both the approaches have advantages and dis-advantages .in content based filtering the it based on the user ratings or user likes only such kind of movie will recommended to the user.

Advantages: it is easy to design and it takes less time to compute

Dis-advantages: the model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

In Collaborative filtering the recommendation is comparison of similar users.

Advantages: No need domain knowledge because the embeddings are automatically learned. The model can help users discover new interests. In isolation, the ML system may not know the user is interested in a given item, but the model might still recommend it because similar users are interested in that item.

Dis-advantages: The prediction of the model for a given (user, item) pair is the dot product of the corresponding embeddings. So, if an item is not seen during training, the system can't create an embedding for it and can't query the model with this item. This issue is often called the **cold-start problem**.

The hybrid approach will resolves all these limitations by combining both content and collaborative filtering

CODE-

```

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel) O



In [1]: import numpy as np  

import pandas as pd



In [2]: movies = pd.read_csv('tmdb_5000_movies.csv')  

credits = pd.read_csv('tmdb_5000_credits.csv')



In [3]: movies.head()



|   | budget    | genres                                                    | homepage                                     | id     | keywords                                                    | original_language | original_title                           | overview                                            | popularity | production_companies                                 |
|---|-----------|-----------------------------------------------------------|----------------------------------------------|--------|-------------------------------------------------------------|-------------------|------------------------------------------|-----------------------------------------------------|------------|------------------------------------------------------|
| 0 | 237000000 | [{"id": 28, "name": "Action"}, {"id": 12, "name": ...]    | http://www.avatarmovie.com/                  | 19995  | [{"id": 1463, "name": "culture clash"}, {"id": ...]         | en                | Avatar                                   | In the 22nd century, a paraplegic Marine is di...   | 150.437577 | [{"name": "Inglourious Basterds", "id": 150.437577}  |
| 1 | 300000000 | [{"id": 12, "name": "Adventure"}, {"id": 14, "name": ...] | http://disney.go.com/disneypictures/pirates/ | 285    | [{"id": 270, "name": "ocean"}, {"id": 726, "name": ...]     | en                | Pirates of the Caribbean: At World's End | Captain Barbosa, long believed to be dead, ha...    | 139.082615 | [{"name": "Walt Disney Pictures", "id": 139.082615}  |
| 2 | 245000000 | [{"id": 28, "name": "Action"}, {"id": 12, "name": ...]    | http://www.sonypictures.com/movies/spectre/  | 206647 | [{"id": 470, "name": "spy"}, {"id": 818, "name": ...]       | en                | Spectre                                  | A cryptic message from Bond's past sends him o...   | 107.376788 | [{"name": "Columbia Pictures", "id": 107.376788}     |
| 3 | 250000000 | [{"id": 28, "name": "Action"}, {"id": 80, "name": ...]    | http://www.thedarkknightrises.com/           | 49026  | [{"id": 849, "name": "dc comics"}, {"id": 853, "name": ...] | en                | The Dark Knight Rises                    | Following the death of District Attorney Harvey...] | 112.312950 | [{"name": "Legends Entertainment", "id": 112.312950} |


```

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel) O

In [9]: movies.isnull().sum()

Out[9]:

movie_id	0
title	0
overview	3
genres	0
keywords	0
cast	0
crew	0
dtype:	int64

In [10]: movies.dropna(inplace=True)

In [11]: movies.duplicated().sum()

Out[11]: 0

In [12]: import ast

In [13]: def convert(text):

L = []

for i in ast.literal_eval(text):

L.append(i['name'])

return L

In [14]: movies['genres'] = movies['genres'].apply(convert)

In [15]: movies.head()

Out[16]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[{"cast_id": 242, "character": "Jake Sully", ...]	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbosa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...]	[{"cast_id": 4, "character": "Captain Jack Spa...]	[{"credit_id": "52fe4232c3a36847f800b5797", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...]	[{"cast_id": 1, "character": "James Bond", "cr...]	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harvey...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...]	[{"cast_id": 2, "character": "Bruce Wayne / Ba...]	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...]	[{"cast_id": 5, "character": "John Carter", "cr...]	[{"credit_id": "52fe479ac3a36847f813ea3", "de...

In [17]: def convert3(text):

L = []

counter = 0

for i in ast.literal_eval(text):

if counter < 3:

L.append(i['name'])

counter+=1

return L

In [18]: movies['cast'] = movies['cast'].apply(convert3)

movies.head()


```


```

```

Out[69]: 'in the 22nd century, a parapleg marin is dispatch to the moon pandora on a uniqu mission, but becom torn between follow order
and protect an alien civilization. action adventur fantasi sciencefict cultureclash futur spacewar spacecoloni societi spacetra
vel futurist romanc space alien tribe alienplanet cgi marin soldier battl loveaffair antiwar powerrel mindandsoul 3d samworthin
gton zoesaldana sigourneyweav jamescameron'

In [70]: from sklearn.metrics.pairwise import cosine_similarity

In [77]: similarity = cosine_similarity(vectors)

In [78]: similarity

Out[78]: array([[1.          , 0.08458258, 0.08718573, ... , 0.04559608, 0.          ,
   0.          ],
   [0.08458258, 1.          , 0.06063391, ... , 0.02378257, 0.          ,
   0.02615329],
   [0.08718573, 0.06063391, 1.          , ... , 0.02451452, 0.          ,
   0.          ],
   ...,
   [0.04559608, 0.02378257, 0.02451452, ... , 1.          , 0.03962144,
   0.04229549],
   [0.          , 0.          , 0.          , ... , 0.03962144, 1.          ,
   0.08714204],
   [0.          , 0.02615329, 0.          , ... , 0.04229549, 0.08714204,
   1.          ]])

In [92]: def recommend(movie):
    movie_index = new_df[new_df['title'] == movies].index[0]
    distances = similarity[movie_index]
    movies_list = sorted(list(enumerate(distances)), reverse=True, key = lambda x: x[1])[1:6]

    for i in movies_list:
        print(new_df.iloc[i[0]].title)

    [0.04559608, 0.02378257, 0.02451452, ... , 1.          , 0.03962144,
    0.04229549],
    [0.          , 0.          , 0.          , ... , 0.03962144, 1.          ,
    0.08714204],
    [0.          , 0.02615329, 0.          , ... , 0.04229549, 0.08714204,
    1.          ]]

In [92]: def recommend(movie):
    movie_index = new_df[new_df['title'] == movies].index[0]
    distances = similarity[movie_index]
    movies_list = sorted(list(enumerate(distances)), reverse=True, key = lambda x: x[1])[1:6]

    for i in movies_list:
        print(new_df.iloc[i[0]].title)

In [98]: recommend('Batman Begins')

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_22144\701328947.py:2: FutureWarning: Automatic reindexing on DataFrame vs Series comparisons is deprecated and will raise ValueError in a future version. Do `left, right = left.align(right, axis=1, copy=False)` before e.g. `left == right`
    movie_index = new_df[new_df['title'] == movies].index[0]

    Aliens vs Predator: Requiem
    Falcon Rising
    Independence Day
    Titan A.E.
    Aliens

In [99]: import pickle

In [97]: pickle.dump(new_df, open('movie_list.pkl', 'wb'))

```

TABLE 1:

PARAMETERS	COLLABORATIVE	CONTENT BASED	PROPOSED
	APPROACH	APPROACH	APPROACH
Accuracy	Low	Average	High
Quality	Low	Average	High
Scalability	Less	Average	High
Computing Time	Average	High	Low
Memory	Average	Low	High

A. Comparison with Existing Technology : The table 2 and 3 compares the result of the proposed system with the existing technique. These tables shows a comparison of RMSE with the existing technique i.e. cuckoo search. It is seen from the tables that for the existing technique the RMSE value is 1.23154 for cluster equal to 68, RMSE value using proposed technique is 1.233 to 19 clusters and RMSE value using proposed technique is 1.081648 to 2 clusters

TABLE II: RMSE in Proposed Technique

Root Mean Squared Error	No. of Cluster
1.23154	68

TABLE III: RMSE in Existing Technique

Root Mean Squared Error	No. of Cluster
1.2333	19
1.081648	2

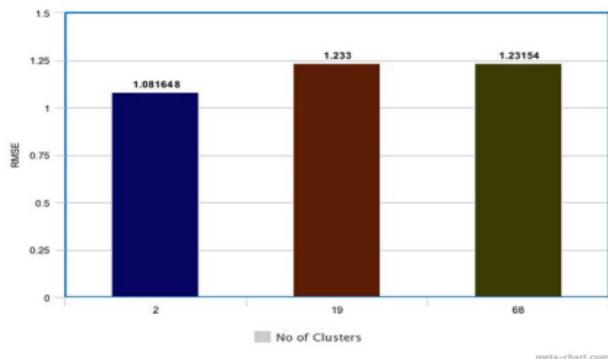


Fig. 9: Comparison Graph with the Existing Technique

CHAPTER 7

TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve: -

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.
- To provide operational reliability of the system.

7.1 Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage. Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process. Test Left is a tool that allows advanced testers and developers to shift left with the fastest test automation tool embedded in any IDE.

Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application

behave as expected (i.e, the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests

System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

In this project, to improve the accuracy, quality and scalability of movie recommendation system, a Hybrid approach by unifying content based filtering and collaborative filtering; using Singular Value Decomposition (SVD) as a classifier and Cosine Similarity is presented in the proposed methodology. Existing pure approaches and proposed hybrid approach is implemented on three different Movie datasets and the results are compared among them. Comparative results depicts that the proposed approach shows an improvement in the accuracy, quality and scalability of the movie recommendation system than the pure approaches. Also, computing time of the proposed approach is lesser than the other two pure approaches.

8.2 Future scope:

In the proposed approach, It has considered Genres of movies but, in future we can also consider age of user as according to the age movie preferences also changes, like for example, during our childhood we like animated movies more as compared to other movies. There is a need to work on the memory requirements of the proposed approach in the future. The proposed approach has been implemented here on different movie datasets only. It can also be implemented on the Film Affinity and Netflix datasets and the performance can be computed in the future.

REFERENCES -

- [1] Goel A., Khandelwal D., Mundhra J., Tiwari R. (2018) Intelligent and Integrated Book Recommendation and Best Price Identifier System Using Machine Learning. In: Bhateja V., Coello Coello C., Satapathy S., Pattnaik P. (eds) Intelligent Engineering Informatics. Advances in Intelligent Systems and Computing, vol 695. Springer, Singapore .
- [2] Bao J., Zheng Y. (2017) Location-Based Recommendation Systems. In: Shekhar S., Xiong H., Zhou X. (eds) Encyclopedia of GIS. Springer, Cham .
- [3] Chavarriaga O., Florian-Gaviria B., Solarte O. (2014) A Recommender System for Students Based on Social Knowledge and Assessment Data of Competences. In: Rensing C., de Freitas S., Ley T., Munoz-Merino ~ P.J. (eds) Open Learning and Teaching in Educational Communities. EC-TEL 2014. Lecture Notes in Computer Science, vol 8719. Springer, Cham .
- [4] F.O.Isinkaye et. al, Recommendation systems: Principles, methods and evaluation, Egyptian Informatics Journal Volume 16, Issue 3, November 2015.
- [5] H. Drachsler, T. Bogers, R. Vuorikari, K. Verbert, E. Duval, N. Manouselis, G. Beham, S. Lindstaedt, H. Stern, M. Friedrich, et al. Issues and considerations regarding sharable data sets for recommender systems in technology enhanced learning. Procedia Computer Science, 1(2):2849–2858, 2010.
- [6] AlvaroTejeda-Lorente, A quality based recommender system to disseminate information in a university digital library,Information Sciences Volume 261, 10 March 2014, Pages 52-69 .
- [7] Trang Tran, T.N., Atas, M., Felfernig, A. et al. J Intell Inf Syst (2018) 50: 501. <https://doi.org/10.1007/s10844-017-0469-0> .
- [8] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu,An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems,IEEE Transactions on Industrial Informatics (Volume: 10 , Issue: 2 , May 2014) .
- [9] Badsha, S., Yi, X. Khalil, I. Data Sci. Eng. (2016) 1: 161. <https://doi.org/10.1007/s41019-016-0020-2> .
- [10] Farman Ullah, Ghulam Sarwar, Sung Chang Lee, Yun Kyung Park, Kyeong Deok Moon, Jin Tae Kim, Hybrid Recommender System with Temporal Information,The International Conference on Information Network, 2012,DOI: 10.1109/ICOIN.2012.6164413.

Java Course By CodeWithHarry

Java is an Object Oriented programming language developed by Sun Microsystems of USA in 1991

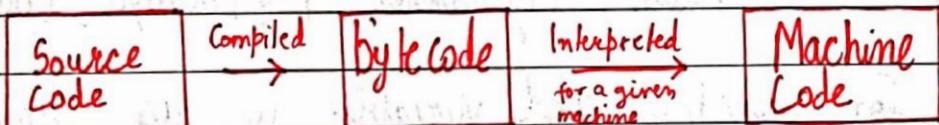
It was originally called Oak by James Gosling

↳ one of the inventors of Java!

JAVA = Purely Object Oriented

How JAVA Works?

Java is compiled into the bytecode and then it is interpreted to machine code



JAVA Installation

Go to Google & type "Install JDK" \Rightarrow Installs JAVA JDK

Go to Google & type "Install IntelliJ Idea" \Rightarrow Installs JAVA IDE

JDK \rightarrow JAVA Development Kit = Collection of tools used for developing and running Java programs

JRE \rightarrow JAVA Runtime Environment = Helps in executing programs developed in JAVA

Basic Structure of a Java Program

```
package com.Company; → Groups classes!  
public class Main { → Entry point into the application  
    public static void main (String [] args) {  
        System.out.println ("Hello World");  
    } }
```

Naming Conventions

→ For classes, we use Pascal Convention. First and Subsequent characters from a word are Capital letters (uppercase)

Example :

Main, MyScanner, MyEmployee, CodeWithHarry

→ For functions and variables, we use CamelCase Convention. Here first character is lowercase and the subsequent characters are uppercase like below:

main, myScanner, myMarks, CodeWithHarry

Chapter 1 - Variables and datatypes

Just like we have some rules that we follow to speak English (the grammar), we have some rules to follow while writing a Java program. The set of these rules is called syntax.

↳ Vocabulary & Grammar of Java

Variables

A variable is a container that stores a value. This value can be changed during the execution of the program.

Example :

int number = 8; *Value it stores!*
Data type *variable name*

Rules for declaring a variable name

We can choose a name while declaring a Java variable if the following rules are followed :

1. Must not begin with a digit → int 1arry; is invalid!
2. Name is case sensitive → harry and Harry are different!
3. Should not be a keyword (like void)
4. White Space not allowed. → int Code With Harry; is invalid
5. Can contain alphabets, \$ character, _ character and digits if the other conditions are met.

Data Types

Data types in Java fall under the following categories

1. Primitive Data Types (Intrinsic)
2. Non-Primitive Data Types (Derived)

Primitive Data Types

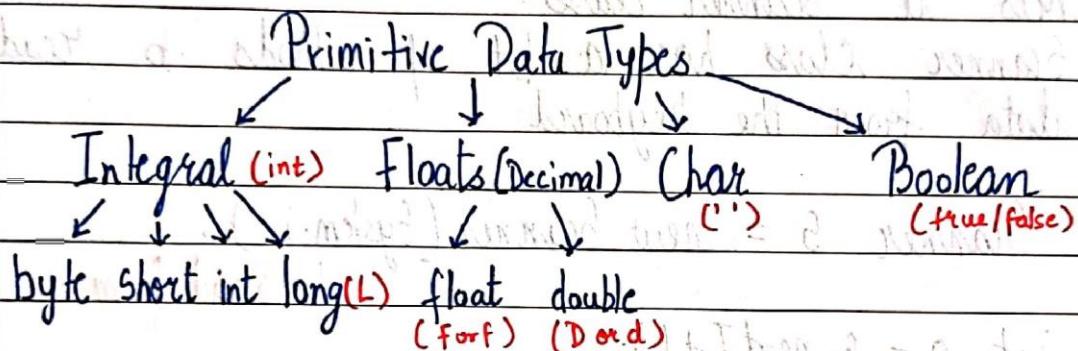
Java is statically typed. → Variables must be declared before use.
There are 8 primitive data types supported by Java:

- 1, byte →
 - Value ranges from -128 to 127
 - Takes 1 byte
 - Default value is 0
- 2, short →
 - Value ranges from $(2^{16})/2$ to $(2^{16})/2 - 1$
 - Takes 2 bytes
 - Default value is 0
- 3, int →
 - Value ranges from $(2^{32})/2$ to $(2^{32})/2 - 1$
 - Takes 4 bytes
 - Default value is 0
- 4, float →
 - Value ranges from (See Docs)
 - Takes 4 bytes
 - Default value is 0.0f
- 5, long →
 - Value ranges from $(2^{64})/2$ to $(2^{64})/2 - 1$
 - Takes 8 bytes
 - Default value is 0
- 6, double →
 - Value ranges from (See docs)
 - Takes 8 bytes
 - Default value is 0.0d
- 7, char →
 - Value ranges from 0 to 65535 ($2^{16} - 1$)
 - Takes 2 bytes → because it supports unicode
 - Default value is '\u0000'

- 8) boolean →
- Value can be true or false
 - Size depends on JVM
 - Default value is false

Quick Quiz : Write a Java program to add three numbers.

How to choose data types for our Variables



In order to choose the data type we first need to find the type of data we want to store. After that we need to analyze the Min & Max value we might use

Literals

A Constant value which can be assigned to the variable is called as a literal

- 101 → Integer literal
- 10.1f → Float literal
- 10.1 → double literal (default type for decimals)
- 'A' → character literal
- true → boolean literal
- "Harry" → String literal

Keywords

Words which are reserved and used by the Java Compiler. They cannot be used as an Identifier.



Go to docs.oracle.com for a comprehensive list!

Reading data from the Keyboard

In order to read data from the Keyboard, Java has a Scanner class.

Scanner class has a lot of methods to read the data from the keyboard

Scanner s = new Scanner(System.in);

↳ Read from the keyboard

int a = s.nextInt();

↳ Method to read from the keyboard

(Integer in this case)

Exercise 1.1

Write a Program to calculate percentage of a given student in CBSE board exam. His marks from 5 subjects must be taken as input from the keyboard. (Marks are out of 100).

Chapter 1 - Practice Set

- 1 Write a program to sum three numbers in Java
- 2 Write a program to calculate CGPA using marks of three subjects (out of 100).
- 3 Write a Java program which asks the user to enter his/her name and greets them with "Hello <name>, have a good day" text.
- 4 Write a Java program to convert Kilometers to miles
- 5 Write a Java program to detect whether a number entered by the user is integer or not.

Chapter 2 - Operators and Expressions

Operators are used to perform operations on variables and values.

$$7 + 11 = 18$$

operand operator operand Result

Types of operators

- Arithmetic Operators → $+, -, *, /, \%, ++, --$
- Assignment operators → $=, +=$
- Comparison operators → $==, >=, <=$
- Logical operators → $\&, ||, !$
- Bitwise Operators → $\&, |$ (operates bitwise)

Arithmetic operators cannot work with booleans
% operator can work on floats & doubles

Precedence of operators

The operators are applied and evaluated based on precedence. For example $(+, -)$ has less precedence compared to $(*, /)$. Hence $* \& /$ are evaluated first.

In case we like to change this order, we use parenthesis

Associativity

Associativity tells the direction of execution of operators. It can either be Left to Right or Right to left

$* / \rightarrow L \text{ to } R$

$+ - \rightarrow L \text{ to } R$

$++, -= \rightarrow R \text{ to } L$

Quick Quiz : How will you write the following expressions in Java ?

$$\frac{x-y}{2}, \frac{b^2-4ac}{2a}, \sqrt{v^2-u^2}, a * b - d$$

Resulting data type after arithmetic operation

following table summarizes the resulting data types after arithmetic operation on them

$$R = b + s \rightarrow \text{int}$$

$b \rightarrow \text{byte}$ $f \rightarrow \text{float}$

$$R = s + i \rightarrow \text{int}$$

$s \rightarrow \text{short}$ $d \rightarrow \text{double}$

$$R = l + f \rightarrow \text{float}$$

$i \rightarrow \text{integer}$ $c \rightarrow \text{character}$

$$R = i + f \rightarrow \text{float}$$

$l \rightarrow \text{long}$

$$R = c + i \rightarrow \text{int}$$

$$R = c + s \rightarrow \text{int}$$

$$R = l + d \rightarrow \text{double}$$

$$R = f + d \rightarrow \text{double}$$

Increment and Decrement Operators

$a++$, $++a \rightarrow$ Increment operators \rightarrow Data type

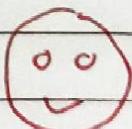
$a--$, $--a \rightarrow$ Decrement operators \rightarrow remains same

These will operate on all data types except booleans

Quick Quiz : Try increment and decrement operators on a Java variable

$a++ \rightarrow$ first use the value and then increment

$++a \rightarrow$ first increment the value then use it



Quick Quiz : What will be the value of the following expression (x).

int y = 7;

int x = ++y * 8;

Value of x ?

char a = 'B';

a++; → a is now 'C'

Chapter 2 - Practice Set

1 What will be the result of the following expression

$$\text{float } a = 7 / 4 * 9 / 2$$

2 Write a java program to encrypt a grade by adding 8 to it. Decrypt it to show the correct grade.

3 Use comparison operators to find out whether a given number is greater than the user entered number or not.

4 Write the following expression in a java program:

$$\frac{v^2 - u^2}{2 \cdot 5}$$

5 find the value of the following expression:

$$\text{int } x = 7$$

$$\text{int } a = 7 * 49 / 7 + 35 / 7$$

Value of a ?

Chapter 3 - Strings

A string is a sequence of characters

A string is instantiated as follows:

```
String name;  
name = new String ("Harry");
```

String is a class but can be used like a data type:

[Strings are immutable and cannot be changed]

```
String name = "Harry";  
Reference                      Object
```

Different ways to print in Java

We can use the following ways to print in Java:

1. `System.out.print()` → No newline at the end!
2. `System.out.println()` → Prints a new line at the end
3. `System.out.printf()`
4. `System.out.format()`

`System.out.printf("%c", ch)`

→ %d for int
%f for float
%c for char
%s for string

String Methods

String methods operate on Java Strings. They can be used to find length of the string, convert to lowercase, etc.

Some of the commonly used String methods are :-

String name = "Harry";
 ^ 1 2 3 4

1. name.length() → Returns length of String name.
(5 in this case)
 2. name.toLowerCase() → Returns a new String which has all the lowercase characters from the String name.
 3. name.toUpperCase() → Returns a new String which has all the uppercase characters from the String name.
 4. name.trim() → Returns a new String after removing all the leading and trailing spaces from the original String.
 5. name.substring(int start) → Returns a substring from start to the end.
substring(3)
Returns "ry"
[Note that index starts from 0]
 6. name.substring(int start, int end) → Returns a substring from start index to the end index. Start index is included and end is excluded
- char char
 ↑ ↑
7. name.replace('r', 'p') → Returns a new string after replacing r with p. Happy is returned in this case.

8. `name.startsWith("Ha")` → returns true if name starts with string "Ha". true in this case!
9. `name.endsWith("ry")` → returns true if name ends with string "ry". true in this case.
10. `name.charAt(2)` → returns character at a given index position. r in this case!
11. `name.indexOf("ar")` → returns the index of the given string.
For ex: `name.indexOf("ar")` returns 1 which is the first occurrence of ar in string "Harry", -1 otherwise
12. `name.indexOf("s", 3)` → returns the index of the given string starting from the index 3 (int). -1 is returned in this case!
13. `name.lastIndexOf("r")` → returns the last index of the given string. 3 in this case!
14. `name.lastIndexOf("r", 2)` → returns the last index of the given string before index 2.
15. `name.equals("Harry")` → returns true if the given string is equal to "Harry" false otherwise [case sensitive]

16 name.equalsIgnoreCase("harry") → returns true if two strings are equal ignoring the case of characters.

Escape Sequence Characters

Sequence of characters after backslash ``

= Escape sequence characters

Escape sequence characters consist of more than one characters but represents one character when used within the strings.

Examples : \n, \t, \', \", etc.

newline Tab singlequote backslash

Chapter 3 - Practice Set

- 1 Write a Java program to convert a string to lowercase
- 2 Write a Java program to replace spaces with underscores.
- 3 Write a Java program to fill in a letter template which looks like below:
`letter = "Dear <|name|>, Thanks a lot"`
Replace <|name|> with a string (some name)
- 4 Write a Java program to detect double and triple spaces in a string
- 5 Write a program to format the following letter using escape sequence characters.
`letter = "Dear Harry, This Java Course is nice. Thanks"`

Chapter 4 - Conditionals in Java

Sometimes we want to watch comedy videos on YouTube if the day is Sunday.

Sometimes, we order junk food if it is our friend's birthday in the hostel.

You might want to buy an Umbrella if its raining and you have the money.

You order the meal if also or your favorite bhindi is listed on the menu.

All these are decisions which depends on a certain condition being met.

In Java, we can execute instructions on a condition being met.

Decision making Instructions in Java

- If - Else Statement
- Switch Statement

If - else Statement

The syntax of an If - Else statement in C looks like that of C++ and JavaScript. Java has a similar syntax too. It looks like:

```
if (Condition - to - be - checked) {  
    Statements - if - Condition - true;  
}
```

```
else {  
    Statements - if - Condition - false;  
}
```

Code Example :

```
int a = 29;
if ( a > 18 ) {
    System.out.println(" You can drive");
}
```

Note that the else block is optional

Relational Operators in Java

Relational operators are used to evaluate conditions (true or false) inside the if statements.

Some examples of relational operators are :

$=$, \geq , $>$, $<$, \leq , \neq

\downarrow equals \downarrow greater than or eq. to \downarrow Not equals

Note : '=' is used for assignment whereas ' $=$ ' is used for equality check.

The condition can be either true or false.

Logical Operators

&&, || and ! are most commonly used logical operators in Java

These are read as :

$\&\&$	\rightarrow	AND
$ $	\rightarrow	OR
$!$	\rightarrow	NOT

\Rightarrow Used to provide logic to our JAVA programs

AND operator

Evaluates to true if both the conditions are true

$$Y \& Y = Y$$

$Y \rightarrow \text{true}$

$$Y \& N = N$$

$N \rightarrow \text{false}$

$$N \& Y = N$$

$$N \& N = N$$

OR Operator

Evaluates to true when at least one of the conditions is true.

$$Y || Y = Y$$

$Y \rightarrow \text{true}$

$$Y || N = Y$$

$N \rightarrow \text{false}$

$$N || Y = Y$$

$$N || N = N$$

NOT Operator

Negates the given logic (true becomes false and false becomes true)

$$! Y = N$$

$Y \rightarrow \text{true}$

$$! N = Y$$

$N \rightarrow \text{false}$

else if clause

Instead of using multiple if statements, we can also use else if along with if thus forming an if-else-if-else ladder

Using such kind of logic reduces indents. last else is executed only if all the conditions fail.

```
if (Condition) {  
    // Statements;  
}  
else if {  
    // Statements;  
}  
else {  
    // Statements;  
}
```

Switch Case Control Instruction

Switch - Case is used when we have to make a choice between number of alternatives for a given variable

```
Switch (Var) {  
    Case C1 :  
        // Code;  
        break;  
    Case C2 :  
        // Code  
        break;  
    Case C3 :  
        // Code  
        break;  
    default :  
        // Code
```

{

Var can be an integer, character or String in Java.

A Switch can occur within another but in practice this is rarely done

Chapter 4 - Practice Set

1 What will be the output of this program:

```
int a = 10;  
if (a == 11)  
    System.out.println("I am 11");  
else  
    System.out.println("I am not 11")
```

2 Write a program to find out whether a student is pass or fail; if it requires total 40% and at least 33% in each subject to pass. Assume 3 subjects and take marks as input from the user.

3 Calculate income tax paid by an employee to the government as per the slabs mentioned below:

Income Slab	Tax
2.5L - 5.0L	5%
5.0L - 10.0L	20%
Above 10.0L	30%

Note that there is no tax below 2.5L. Take input amount as an input from the user.

4 Write a Java program to find out the day of the week given the number [1 for Monday, 2 for Tuesday ... and so on!]

5 Write a Java program to find whether a year entered by the user is a leap year or not.

6 Write a program to find out the type of website from the URL

- com → Commercial website
- org → organization website
- in → Indian website

Chapter 5 - Loop Control Instruction

Sometimes we want our programs to execute a few set of instructions over and over again for example - print 1 to 1000, print multiplication table of 7, etc.

Loops make it easy for us to tell the computer that a given set of instructions need to be executed repeatedly.

Types of Loops

Primarily, there are three types of loops in Java:

- 1, While loop
- 2, do-while loop
- 3, for loop

We will look into these one by one.

While loops

While (boolean condition)
 {

 // Statement
 }

→ This keeps executing as long as the condition is true.

If the condition never becomes false, the while loop keeps getting executed. Such a loop is known as an infinite loop.

Quick Quiz : Write a program to print natural numbers from 100 to 200.

do - while loop

This loop is similar to a while loop except the fact that it is guaranteed to execute at least once.

do {

// Code

} while (condition);

→ Note this Semicolon

while → checks the condition & executes the code

do - while → Executes the code & then checks the condition

Quick Quiz : Write a program to print first n natural numbers using do - while loop.

for Loop

The syntax of a for loop looks like this :

for (initialize; check bool expression; update) {

// Code ;

}

A for loop is usually used to execute a piece of code for specific number of times.

Quick Quiz : Write a program to print first n odd numbers using a for loop.

Decrementing for loop

```
for( i= 7 ; i != 0 ; i--){  
    System.out.println(i);  
}
```

This for loop keeps running until i becomes 0.

Quick Quiz : Write a program to print first n natural numbers in reverse order

break statement

The break statement is used to exit the loop irrespective of whether the condition is true or false.

Whenever a "break" is encountered inside the loop, the control is sent outside the loop.

Continue statement

The continue statement is used to immediately move to the next iteration of the loop.

The control is taken to the next iteration thus skipping everything below "Continue" inside the loop for that iteration.

In A Nut Shell...

1. break statement completely exits the loop
2. Continue statement skips the particular iteration of the loop.

Chapter 5 - Practice Set

1 Write a program to print the following pattern

* * * *
* * *
* *
*

2 Write a program to sum first n even numbers using while loop.

3 Write a program to print multiplication table of a given number n.

4 Write a program to print multiplication table of 10 in reverse order.

5 Write a program to find factorial of a given number using for loops.

6 Repeat 5 using while loop

7 Repeat 1 using for /while loop

8 What can be done using one type of loop can also be done using the other two types of loops - True or False.

9 Write a program to calculate the sum of the numbers occurring in the multiplication table of 8.

10 A do while loop is executed :- 3 methods

- 1> At least once
- 2> At least twice
- 3> At most once

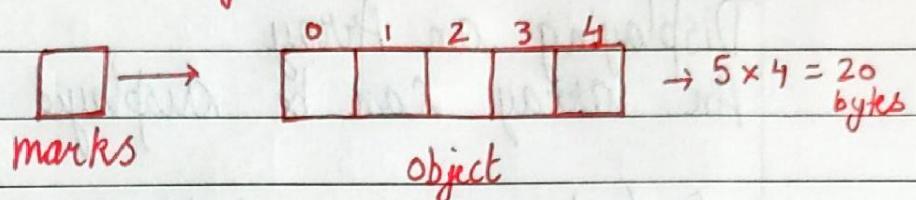
11 Repeat 2 using for loop.

Chapter 6 - Arrays

Array is a collection of similar types of data

Use Case : Storing marks of 5 Students

`int [] marks = new int [5] \Rightarrow [data Type ArrName];`



Accessing Array Elements

Array elements can be accessed as follows

`marks[0] = 100`

`marks[1] = 70`

\vdots

`marks[4] = 98`

\Rightarrow Note that index starts from 0

So in a nut shell, this is how array works :

1. `int [] marks;` \rightarrow Declaration !

`marks = new int [5];` \rightarrow Memory Allocation !

2. `int [] marks = new int [5];` \rightarrow Declaration + Memory Allocation !

3. `int [] marks = {100, 70, 80, 71, 98};` \rightarrow Declare + Initialize !

Array indices starts from 0 and goes till $(n-1)$
where n is the size of the array.

Array length

Arrays have a length property which gives the length of the array

`marks.length` → gives 5 if marks is a reference to array with 5 elements

Displaying an Array

An array can be displayed using a for loop:

```
for (int i=0 ; i < marks.length ; i++)
```

```
{     System.out.println(marks[i]); }
```

⇒ Array Traversal

Quick Quiz: Write a Java program to print the elements of an array in reverse order.

for-each loop in Java

Array elements can also be traversed as follows:

```
for (int element : Arr) {
```

```
    System.out.println(element); }
```

⇒ Prints all the elements

Multidimensional Arrays

Multidimensional Arrays are Array of Arrays

Each element of a M-D array is an array itself
marks in the previous example was a 1-D array.

Multidimensional 2-D Array

A 2-D array can be created as follows:

```
int [][] flats = new int [2][3]
```

↳ A 2-D array of 2 rows + 3 columns

We can add elements to this array as follows

flats [0][0] = 100

flats [0][1] = 101

flats [0][2] = 102

:

& so on!

This 2-D array can be visualised as follows:

	[0]	[1]	[2]	
	Col 1	Col 2	Col 3	
[0]	Row 1	(0,0)	(0,1)	(0,2)
[1]	Row 2	(1,0)	(1,1)	(1,2)

Similarly a 3-D array can be created as follows:

```
String [[[ ]]] arr = new String [2][3][4]
```

Chapter 6 - Practice Set

- = 1 Create an array of 5 floats and calculate their sum.
- = 2 Write a program to find out whether a given integer is present in an array or not.
- = 3 Calculate the average marks from an array containing marks of all students in Physics using for-each loop.
- = 4 Create a Java program to add two matrices of size 2×3 .
- = 5 Write a Java program to reverse an array.
- = 6 Write a Java program to find the maximum element in an array.
- = 7 Write a Java program to find the minimum element in a Java array.
- = 8 Write a Java program to find whether an array is sorted or not.

Chapter 7 - Methods in Java

Sometimes our program grows in size and we want to separate the logic of main method to other methods

for instance - If we are calculating average of a number pair 5 times, we can use methods to avoid repeating the logic.

→ DRY = Don't Repeat Yourself

Syntax of a Method

A method is a function written inside a class. Since Java is an Object Oriented language, we need to write the method inside some class

```
dataType name () {  
    // Method body  
}
```

Following method returns sum of two numbers

→ Return type
int mySum(int a, int b) {
 int c = a + b;
 return c; → Return value
}

Calling a Method

A method can be called by creating an object of the class in which the method exists followed by the method call:

Calc obj = new Calc(); → Object Creation
obj.mySum(a, b); → Method call upon an object

The values from the method call (a and b) are copied to the a and b of the function mySum. Thus even if we modify the values a and b inside the method, the values in the main method will not change.

Void return type

When we don't want our method to return anything, we use Void as the return type.

Static keyword

Static keyword is used to associate a method of a given class with the class rather than the object. Static method in a class is shared by all the objects.

Process of method invocation in Java

Consider the method sum:

```
int sum (int a, int b)
{
    return a+b;
}
```

The method is called like this:

```
Calc obj = new Calc();
c = obj.sum(2, 3)
```

The values 2 and 3 are copied to a and b and then $a+b=2+3=5$ is returned in c which is an integer.

Note: In case of Arrays, the reference is passed. Same is the case for Object passing to methods.

Method Overloading

Two or more methods can have same name but different parameters. Such methods are called Overloaded methods.

Void foo()

Void foo (int a)

int foo (int a, int b)

⇒ Overloaded function foo

Method overloading cannot be performed by changing the return type of methods

Variable Arguments (Varargs)

A function with Vararg can be created in Java using the following Syntax:

public static void foo (int ... arr)
{

// arr is available here as int [] arr
}

foo can be called with Zero or more arguments like this :

foo(7) foo(7,8,9) foo(1,2,7,8,9)

We can also create a function bar like this

public static void bar (int a , int arr)
{

// Code
}

→ Atleast one integer is required now

bar can be called as bar(1), bar(1,2), bar(1,7,9,11)
etc.

Recursion

A function in Java can call itself. Such calling of function by itself is called recursion.

Example: Factorial of a number

$$\text{factorial}(n) = n * \text{factorial}(n-1)$$

$\forall n \geq 1$

Quick Quiz: Write a program to calculate (recursion must be used) factorial of a number in Java?

Chapter 7 - Practice Set

1 Write a Java method to print multiplication table of a number n.

2 Write a program using functions to print the following pattern:

*
* *
* * *
* * * *

3 Write a recursive function to calculate sum of first n natural numbers

4 Write a function to print the following pattern

* * * *
* * *
* *
*

5 Write a function to print n^{th} term of fibonacci series using recursion.

6 Write a function to find average of a set of numbers passed as arguments

7 Repeat 4 using Recursion

8 Repeat 2 using Recursion

- 9 Write a function to convert Celsius temperature into fahrenheit.
- 10 Repeat 3 using iterative approach.

Chapter - 8 : Introduction to OOPs

Object Oriented programming tries to map code instructions with real world making the code short and easier to understand.

What is Object Oriented Programming?

Solving a problem by creating objects is one of the most popular approaches in programming. This is called Object Oriented Programming.

What is DRY?

DRY stands for - Do not repeat yourself

↳ Focuses on code reusability

Class

A class is a blueprint for creating objects.

JEE
Application
Form

⇒ Filled by an Student

Application for
that Student

Class

⇒ Object Instantiation ⇒ Object

Contains info to
create a valid
object.

Object

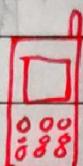
An Object is an instantiation of a class. When a class is defined, a template (info) is defined. Memory is allocated only after object instantiation.

How to model a problem in OOPs
We identify the following:

Noun → Class → Employee
 Adjective → Attributes → name, age, Salary
 Verb → Methods → getSalary(), increment()

OOPs Terminology

1. Abstraction → Hiding internal details [show only essential info!]



⇒ Use this phone without bothering about how it was made

2. Encapsulation → The act of putting various components together (in a capsule).



⇒ Laptop is a single entity with WiFi + Speaker + Storage in a single box!

In Java, encapsulation simply means that the sensitive data can be hidden from the users

3. Inheritance → The act of deriving new things from existing things.

Rickshaw ⇒ E-Rickshaw

Phone ⇒ Smart Phone

Implements DRY!

4. Polymorphism → One entity many forms

Smartphone → Phone

Smartphone → Calculator

Writing a Custom Class

We can write a custom class as follows:

```
public class Employee {  
    int id;           → Attribute 1  
    String name;     → Attribute 2  
}
```

Any real world Object = Properties + Behaviour
Object in OOPs = Attributes + Methods.

A class with Methods

We can add methods to our class Employee as follows:

```
public class Employee {  
    public int id;  
    public String name;  
  
    public int getSalary() {  
        // Code  
    }  
  
    public void getDetails() {  
        // Code  
    }  
};
```

Chapter 8 - Practice Set

- 1 Create a class Employee with following properties and methods:
 - Salary (property) (int)
 - getSalary (method returning int)
 - name (property) (String)
 - getName (method returning String)
 - setName (method changing name)
- 2 Create a class cellphone with methods to print "ringing...", "vibrating..." etc.
- 3 Create a class Square with a method to initialize its side, calculating area, perimeter etc.
- 4 Create a class Rectangle & repeat 3
- 5 Create a class TommyVecetti for Rockstar Games Capable of hitting (print hitting...), running, firing etc.
- 6 Repeat 4 for a Circle.

Chapter 9 - Access Modifiers & Constructors

Access Modifiers

Specifier where a property / method is accessible

There are four types of access modifiers in Java :

- 1> Private
- 2> Default
- 3> Protected
- 4> Public

Getters and Setters

Getter → Returns the value [accessors]

Setter → Sets/Updates the value [mutators]

Example :

```
public class Employee {  
    private int id;  
    private String name;
```

```
    public String getName() {  
        return name;  
    }
```

```
    public void setName() {  
        this.name = "Your-name";  
    }
```

```
    public void setName(String n) {  
        this.name = n;  
    }
```

Quick Quiz : Use these getters and setters from the main method.

Constructors in Java

A member function used to initialize an object while creating it.

```
Employee harry = new Employee();
harry.setName("Harry Bhai");
```

In order to write our own constructor, we define a method with name same as class name.

```
public Employee() {
    name = "Your Name";
}
```

Constructor Overloading in Java

Constructors can be overloaded just like other methods in Java. We can overload the Employee constructor like below :

```
public Employee(String n) {
    name = n;
}
```

Note : ① Constructors can take parameters without being overloaded.

② There can be more than two overloaded constructors

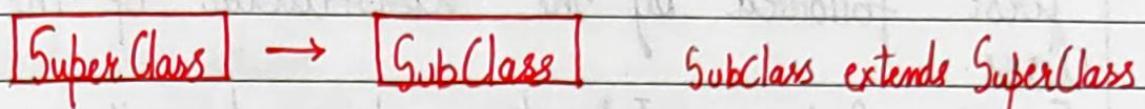
Quick Quiz : Overload the Employee constructor to initialize the Salary to Rs 10,000.

Chapter 9 - Practice Set

- 1 Create a class Cylinder and use Getters and Setters to set its radius and height.
- 2 Use ① to calculate surface area and Volume of the cylinder.
- 3 Use a constructor and repeat ①
- 4 Overload a constructor used to initialize a rectangle of length 4 and breadth 5 for using custom parameters.
- 5 Repeat ① for a sphere

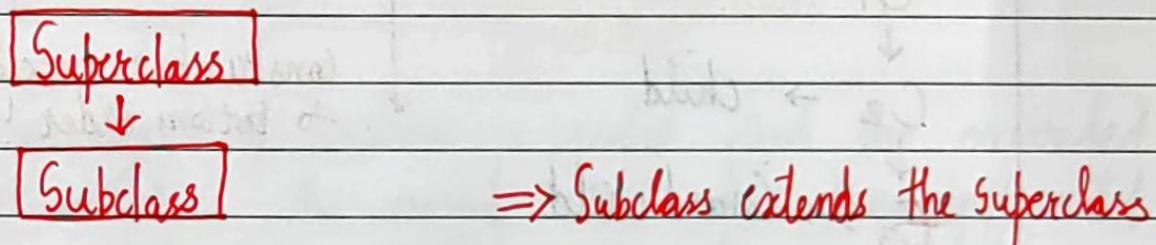
Chapter 10 - Inheritance

Inheritance is used to borrow properties & methods from an existing class

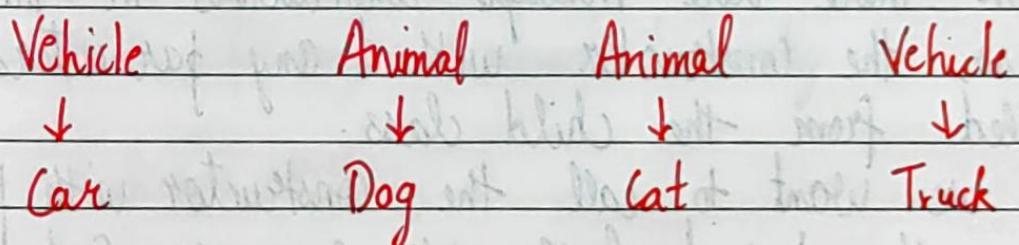


Declaring Inheritance in Java

Inheritance in Java is declared using extends keyword



More Examples



When a class inherits from a superclass, it inherits parts of superclass methods and fields.

Java doesn't support multiple inheritance ie two classes cannot be super classes for a subclass.

Code Example

Inheritance in Java is declared using extends keyword

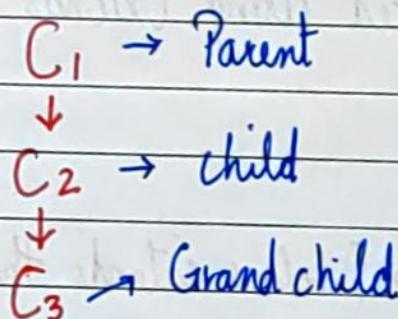
public class Dog extends Animal {→ Inheriting Dog from Animal Class!!}
// Code

Quick Quiz : Create a class Animal and Derive another class Dog from it.

Constructors in Inheritance

When a Derived class is extended from the Base class, the constructor of the Base class is executed first followed by the constructor of the derived class.

For the following Inheritance hierarchy, the constructors are executed in the order ① → ② → ③



↓ Constructors execute in top to bottom order !

Constructors during Constructor Overloading

When there are multiple constructors in the parent class, the constructor without any parameters is called from the child class.

If we want to call the constructor with parameters from the parent class, we can use Super keyword

`Super (a, b);` → Calls the constructor from the parent class which takes 2 variables

this keyword

this is a way for us to reference an object of the class which is being created/referenced.

`this.area = 2` → this is a reference to current object

Super Keyword

A reference variable used to refer immediate parent class object

- Can be used to refer immediate parent class instance variable
- Can be used to invoke parent class methods.
- Can be used to invoke parent class constructors.

Method Overriding

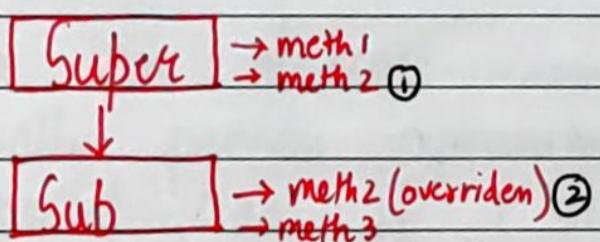
If the child class implements the same method present in the parent class again, it is known as method overriding

↳ Redefining method of super class!
(in sub class)

When an object of subclass is created and the overridden method is called, the method which has been implemented in the subclass is called & its code is executed.

Dynamic method dispatch

Consider the following inheritance hierarchy



Scenario 1 → Super obj = new Sub() → Allowed ✓
 obj. meth2() → ② is called (Method of object)
 obj. meth3() → Not Allowed ✗

Scenario 2 → Sub obj = new Super() → Not Allowed ✗

This is known as Dynamic method dispatch and is used to achieve run time polymorphism in Java.

Chapter 10 - Practice Set

- 1 Create a class Circle and use inheritance to create another class Cylinder from it.
- 2 Create a class Rectangle And use inheritance to create another class Cuboid . Try to keep it as close to real world scenario as possible.
- 3 Create methods for area and Volume in ①
- 4 Create methods for area & volume in ②. Also create getters and setters
- 5 What is the order of constructor execution for the following inheritance hierarchy:

Base



Derived 1



Derived 2

Derived 2 Obj = new Derived2();

Which constructor(s) will be executed & in what order?

Chapter 11 - Abstract Classes & Interfaces

What does Abstract (class) mean?

Abstract in english means → existing in thought or as an idea without concrete existence

Abstract method

A method that is declared without an implementation

```
abstract void moveTo(double x, double y)
```

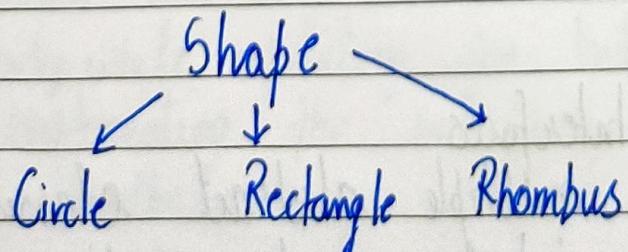
Abstract Class

If a class includes abstract methods, then the class itself must be declared abstract, as in:

```
public abstract class PhoneModel {  
    abstract void switchoff();  
    // more code  
}
```

When an abstract class is subclassed, the subclass usually provides implementations for all of the methods in parent class. If it doesn't, it must be declared abstract.

An Example

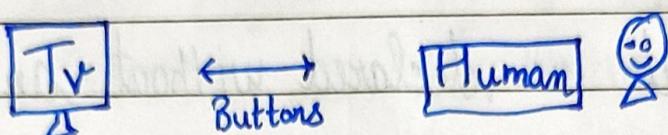


Note - It is possible to create reference of an abstract class
It is not possible to create an object of an abstract class

We can also assign reference of an abstract class to the object of a concrete subclass.

Interfaces in Java

Interface in English is a point where two systems meet and interact



In Java interface is a group of related methods with empty bodies

An Example

```
interface Bicycle {  
    void applyBrake(int decrement);  
    void speedUp(int increment);  
}
```

```
class AvonCycle implements Bicycle {  
    int speed = 7;  
    void applyBrake(int decrement) {  
        speed = speed - decrement;  
    }  
    void speedUp(int increment) {  
        speed = speed + increment;  
    }  
}
```

Abstract class vs Interfaces

We can't extend multiple abstract classes but we can implement multiple interfaces at a time.
Interfaces are meant for dynamic method dispatch

and run time polymorphism

Is multiple inheritance allowed in Java?

Multiple inheritance face problems when there exist methods with same signature in both the super classes.

Due to such problems, Java does not support multiple inheritance directly but the similar concept can be achieved using Interfaces.

A class can implement multiple Interfaces and extend a class at the same time.

- Note :
- ① Interfaces in Java is a bit like the Class but with a significant difference.
 - ② An Interface can only have method signatures, ^{constant} fields and default methods.
 - ③ The class implementing an Interface needs to only declare the methods (not fields).
 - ④ You can create a reference of Interfaces but not the Object.
 - ⑤ Interface methods are public by default.

Default methods

An interface can have static and default methods.

Default methods enable us to add new functionality to existing Interfaces.

This feature was introduced in Java 8 to ensure backward compatibility while updating an Interface.

Classes implementing the interface need not implement the default methods.

Interfaces can also include private methods for default methods to use.

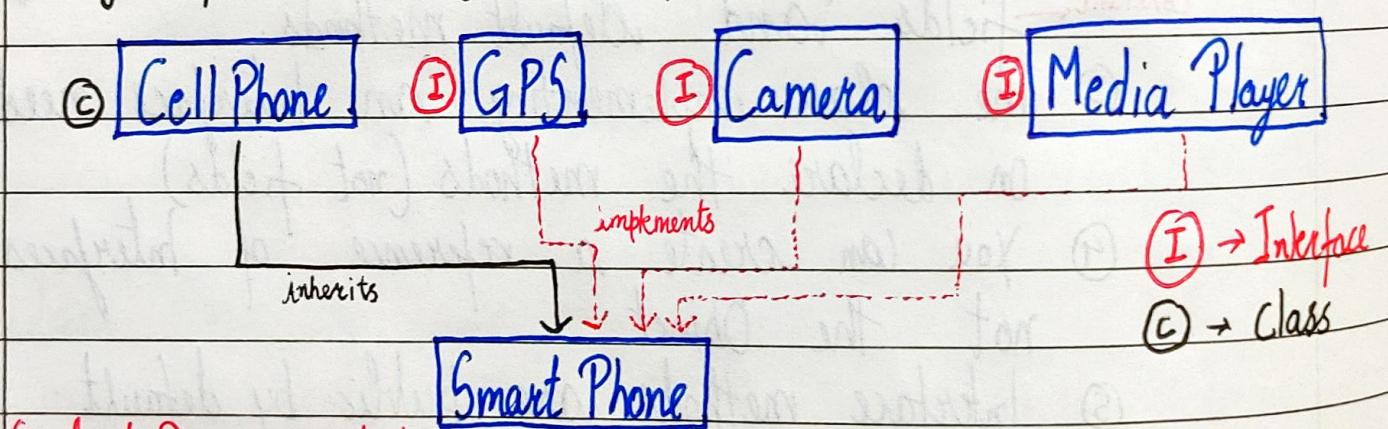
Inheritance in Interfaces
Interfaces can extend another interfaces :

```
public interface Interface1 {
    void meth1();
}
```

```
public interface Interface2 extends Interface1 {
    void meth2();
}
```

Remember that interface cannot implement another interface, only classes can do that!

Polymorphism using Interfaces



Similar to Dynamic method dispatch in Inheritance

GPS g = new SmartPhone(); → Can only use GPS methods
SmartPhone s = new SmartPhone(); → Can only use SmartPhone methods

Implementing an Interface forces method implementation.

Chapter 11 - Practise Set

- 1 Create an abstract class Pen with methods write() and refill() as abstract methods
- 2 Use the Pen class from Q1 to create a concrete class FountainPen with additional method changeNib()
- 3 Create a class Monkey with jump() and bite() methods. Create a class Human which inherits this Monkey class and implements BasicAnimal interface with eat() and sleep methods.
- 4 Create a class Telephone with ring(), lift() and disconnect() methods as abstract methods. Create another class SmartTelephone and demonstrate polymorphism
- 5 Demonstrate polymorphism using monkey class from Ques3.
- 6 Create an Interface TVRemote and use it to inherit another Interface SmartTVRemote.
- 7 Create a class Tv which implements TVRemote interface from Q6

Chapter 12 - Packages

Interpreter vs Compiler

Interpreter translates one statement at a time into machine code.

Compiler scans the entire program and translates whole of it into machine code.

Interpreter



- * One statement at a time
- * Interpreter is needed everytime
- * Partial execution if error
- * Easy for programmers

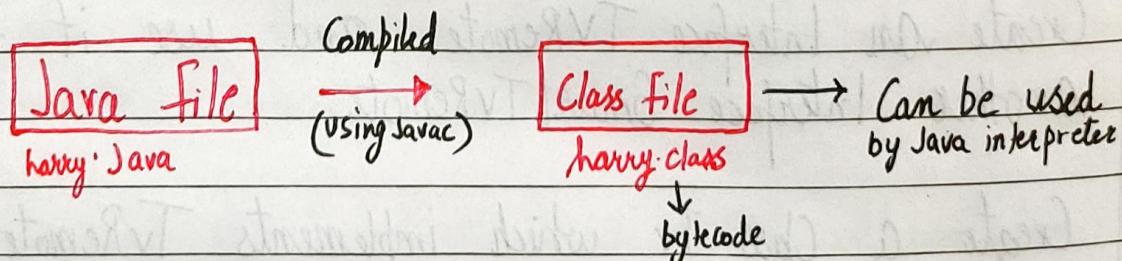
Compiler



- * Entire program at a time
- * Once compiled it is not needed
- * No execution if an error occurs
- * Usually not as easy as Interpreted ones

Is Java Compiled or Interpreted?

Java is a hybrid language → both compiled as well as interpreted



- A JVM can be used to Interpret this bytecode
- This bytecode can be taken to any platform (Win/Mac/Linux) for execution
- Hence Java is platform independent (write once run everywhere)

Executing a Java Program

Java Harry.java → Compiled
java Harry.class → Interpreted

So far the execution of our program was being managed by IntelliJ Idea.
We can download a source code editor like VS Code to compile & execute our Java programs.

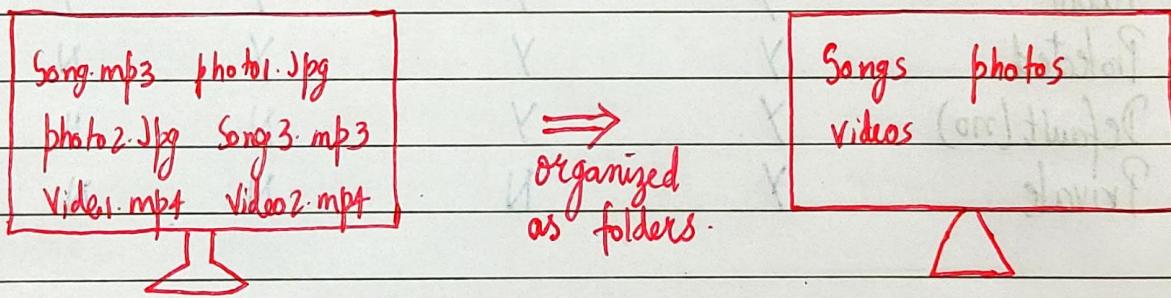
Packages in Java

A package is used to group related classes.

Packages help in avoiding name conflicts.

There are two types of packages:

- * Built in packages → Java API
- * User defined packages → Custom packages



1. class this.java my.mp3
Song.java harry.java
⇒ organized as packages

Using a Java package

import java.lang.* → import everything from java.lang
import java.lang.String → import String from java.lang
s = new java.lang.String("Harry") → Use without importing

Creating a package

javac Harry.java → creates Harry class

javac -d Harry.java → creates a package folder

↳ We can keep adding classes to a package like this

We can also create inner packages by adding "package.inner" as package name

These packages once created can be used by other classes.

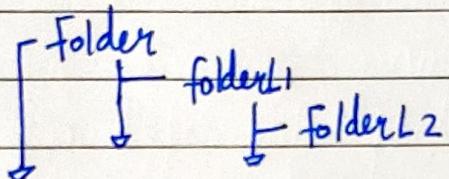
Access Modifiers in Java

Access modifiers determine whether other classes can use a particular field or invoke a particular method. Can be public, private, protected or default (no modifier)

Modifier	Class	Package	Subclass	World
Public	Y	Y	Y	Y
Protected	Y	Y	Y	N
Default (no)	Y	Y	N	N
Private	Y	N	N	N

Chapter 12 - Practice Set

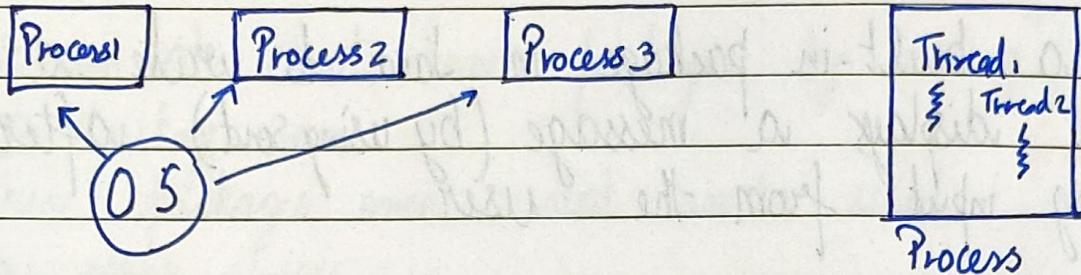
- = 1 Create three classes Calculator, ScCalculator and HybridCalculator and group them into a package.
- = 2 Use a built-in package in Java to write a class which displays a message (by using `System.out`) after taking input from the user.
- = 3 Create a package in class with three package levels folder, folderL1, folderL2



- = 4 Prove that you cannot access default property but can access protected property from the subclass.

Chapter 13 - Multithreading

Multiprocessing and multithreading both are used to achieve multitasking



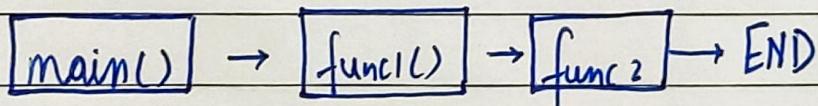
In a nut Shell ..

- Threads use shared memory area
- Threads ⇒ Faster Context switching
- A Thread is light-weight whereas a process is heavyweight

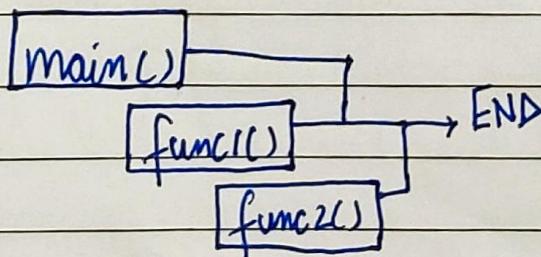
For Example → A word processor can have one thread running in foreground as an editor and another in the background auto saving the document !

Flow of Control in Java

1. Without threading :



2. With threading:



Creating a Thread

There are two ways to create a thread in Java.

1. By extending Thread class
2. By implementing Runnable interface

Life cycle of a Thread

① [New]



② [Runnable]

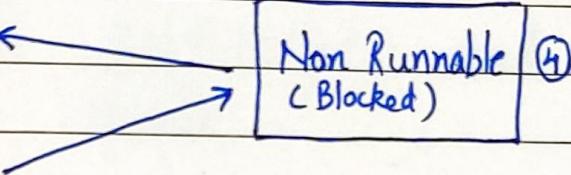


③ [Running]



⑤ [Terminated]

Non Runnable
(Blocked) ④



① New → Instance of thread created which is not yet started by invoking start()

② Runnable → After invocation of start() & before it is selected to be run by the scheduler.

③ Running → After thread scheduler has selected it.

④ Non Runnable → Thread alive, not eligible to run.

⑤ Terminated → run() method has exited

The Thread class

Below are the commonly used Constructors of Thread class:

① Thread()

② Thread(String name)

③ Thread(Runnable r)

④ Thread(Runnable r, String name)

Methods of Thread class

Thread class offers a lot of methods such as run(), start(), join(), getPriority(), setPriority() etc. More can be found on visiting Java docs.

Chapter 13 - Practise Set

- 1 Write a program to print "good morning" and "welcome" continuously on the screen in Java using Threads.
- 2 Add a sleep method in welcome thread of question 1 to delay its execution for 200 ms.
- 3 Demonstrate `getPriority()` and `SetPriority()` methods in Java Threads.
- 4 How do you get state of a given thread in Java?
- 5 How do you get reference to the current thread in Java?

Chapter 14 - Errors & Exceptions

No matter how smart we are, errors are our constant companions. With practice, we keep getting better at finding & correcting them.

There are three types of errors in Java.

- 1> Syntax errors
- 2> Logical errors
- 3> Runtime errors → Also called Exceptions!

Syntax Errors

When compiler finds something wrong with our program, it throws a syntax error.

~~int a = 9 ;
a = a + 3;~~ → No semicolon, Syntax error!

~~d = 4;~~ → Variable not declared, Syntax error!

Logical errors

A logical error or a bug occurs when a program compiles and runs but does the wrong thing.

- message delivered wrongly
- wrong time of chats being displayed
- incorrect redirects!

Runtime Errors

Java may sometimes encounter an error while the program is running. These are also called exceptions!

These are encountered due to circumstances like bad input and (or) resource constraints.

Ex:- user supplies '5+8' to a program which adds 2 numbers.

Syntax errors and logical errors are encountered by the programmer whereas Runtime errors are encountered by the users.

Exceptions in Java

An Exception is an event that occurs when a program is executed disrupting the normal flow of instructions.

There are mainly two types of exceptions in Java:

- 1> Checked Exception → Compile time exceptions (Handled by compiler)
- 2> Unchecked Exception → Runtime exceptions

Commonly Occurring Exceptions

Following are few commonly occurring exceptions in Java:

- 1> NullPointerException
- 2> ArithmeticException
- 3> ArrayIndexOutOfBoundsException
- 4> IllegalArgumentException
- 5> NumberFormatException

try-catch block in Java

In Java, exceptions are managed using try-catch blocks

Syntax:

```
try {  
    // Code to try  
}  
catch (Exception e) {  
    // Code if exception  
}
```

Handling Specific Exceptions

In Java, we can handle specific exceptions by typing multiple catch blocks.

```
try {  
    // Code  
}
```

```
Catch (IOException e) {  
    // Code  
}
```

→ Handles all Exceptions of type IOException

```
Catch (ArithmeticException e) {  
    // Code  
}
```

→ Handles all Exceptions of type ArithmeticException

```
Catch (Exception e) {  
    // Code  
}
```

→ Handles all other Exceptions

Nested try-catch

We can nest multiple try-catch blocks as follows:

```
try {  
    try {  
        // Code  
    }
```

```
    Catch (Ex... c) {  
        // Code  
    }
```

```
    Catch (Ex... c) {  
        // Code  
    }
```

⇒ Nested try-catch blocks

Similarly, we can further nest try catch blocks inside the nested try catch blocks.

Quick Quiz : Write a Java program that allows you to keep accessing an array until a valid index is given by the user.

Exception class in Java

We can write our custom Exceptions using Exception class in Java.

```
public class MyException extends Exception {
    // No overridden methods
}
```

The Exception class has following important methods :

- (1) String `toString()` → executed when `sout(e)` is run
- (2) Void `printStackTrace()` → prints stack trace
- (3) String `getMessage()` → prints the exception message

The Throw keyword

The throw keyword is used to throw an exception explicitly by the programmer

```
if (b == 0) {
    throw new ArithmeticException ("Div by 0");
}
else {
    return a/b;
}
```

In a similar manner, we can throw user defined exceptions :

```
throw new MyException ("Exception thrown");
```

The throws exception

The Java throws keyword is used to declare an exception. This gives an information to the programmer that there might be an exception so it's better to be prepared with a try catch block!

```
public void calculate(int a, int b) throws IOException {  
    // Code  
}
```

Java finally block

finally block contains the code which is always executed whether the exception is handled or not. It is used to execute code containing instructions to release the system resources, close a connection etc.

Chapter 14 - Practice Set

- 1 Write a Java program to demonstrate syntax, logical & runtime errors.
- 2 Write a Java program that prints "HaHa" during Arithmetic exception and "HeHe" during an Illegal argument exception
- 3 Write a program that allows you to keep accessing an array until a valid index is given. If max retries exceed 5 print "Error".
- 4 Modify program in Q3 to throw a custom Exception if max retries are reached.
- 5 Wrap the program in Q3 inside a method which throws your custom Exception.

Advanced Java - 1

Collections Framework

A Collection represents a group of objects.
Java Collections provide Classes and Interfaces
for us to be able to write code quickly and
efficiently.

Why do we need Collections

We need Collections for efficient storage and
better manipulation of data in Java.

For ex: We use arrays to store integers but
what if we want to

- Resize this array ?
- Insert an element in between ?
- Delete an element in Array ?
- Apply certain operations to change
this array ?

How are collections available

Collections in Java are available as Classes and
Interfaces. Following are few commonly used
Collections in Java:

- * ArrayList → For variable size Collection
- * Set → For distinct collection
- * Stack → A LIFO data structure
- * HashMap → For storing key-value pairs

Collection class is available in java.util package

Collection class also provides static methods for
Sorting, Searching etc.