IT Interview Guide For Freshers

Crack your IT interview with confidence

SAMEER S. PARADKAR

bpb

IT Interview Guide

for

Freshers

by

Sameer S. Paradkar

FIRST EDITION 2019

ISBN: 978-93-88511-599

Distributors:

BPB PUBLICATIONS

20, Ansari Road, Darya Ganj

New Delhi-110002

Ph: 23254990/23254991

MICRO MEDIA

Shop No. 5, Mahendra Chambers,

150 DN Rd. Next to Capital Cinema,

V.T. (C.S.T.) Station, MUMBAI-400 001

Ph: 22078296/22078297

DECCAN AGENCIES

4-3-329, Bank Street,

Hyderabad-500195

Ph: 24756967/24756400

BPB BOOK CENTRE

376 Old Lajpat Rai Market,

Delhi-110006

Ph: 23861747

Published by Manish Jain for BPB Publications, 20 Ansari Road, Darya Ganj, New Delhi-110002 and Printed by him at Repro India Ltd, Mumbai

Dedicated to

My parents, wife, kids and all my family members and

friends for their encouragement, support and love.

About the Author

Sameer S. Paradkar is an enterprise architect with 15+ years of solid experience in the ICT industry, which spans across consulting, product development and systems integration. He is an Open Group TOGAF, Oracle Master Java EA, TMForum NGOSS, IBM SOA Solutions, IBM Cloud Solutions, IBM Mobile First, ITIL Foundation V3, COBIT 5 and AWS Solution Architect – Associate certified enterprise architect. He serves as an advisory architect on enterprise architecture programs and continues to work as a subject matter expert. He has worked on multiple architecture transformations and modernization engagements in the USA, UK, Europe, Asia Pacific and the Middle East Regions that presented a phased roadmap to the transformation that maximized the business value while minimizing costs and risks.

Sameer is part of the Architecture Group in AtoS. Prior to AtoS, he has worked in organizations such asEY - IT Advisory, IBM GBS, Wipro Consulting Services, TechMahindra and Infosys Technologies and specializes in IT Strategies and enterprise transformation engagements.

Acknowledgement

Throughout my career, many people have directly and indirectly contributed to this book. I would like to take this opportunity to acknowledge their contribution, influence, and inspiration. I consider myself lucky to have had the opportunity to work with extremely talented and exceptional individuals who extended their whole-hearted support for my work andinitiatives. My eternal thanks to them for believing in me and providing exciting opportunities. I would like to thank my discussion partners, reviewers and supporters, whose valuable comments and feedback have greatly contributed to this book. I look forward to your feedback and valuable inputs on an on-going basis. I would like to thank my current and former colleagues who made my corporate journey exciting, enriching and fulfilling.

– Sameer S Paradkar

Preface

This book is divided into numerous chapters, including the topics that deal with various aspects and stages of the entire interview process. It presents an exhaustive question bank with special emphasis on practical scenarios and business cases. The title describes the qualities that an employer looks for in a potential employee and helps improve the aspirant’s understanding of the interview process. The book begins with providing you top Data Structures and OS sample interview questions, which are asked by the top IT companies while hiring. We will later dive into the concepts and principles of Object-oriented programming. We’ll later cover C/C++/Java/ Programming Interview Questions along with Database Management Systems. Later, you will be taken through the methodologies andprocesses of validations and testing along with CMMI, DevOps, Agile, Scrum, APIs, Micro-services and SOA technologies. Summing it up, towards the end, we will have a set of HR process interview questions, which will cover the best practices to answer interview questions.

To summarize, the key differentiators for the book are:

This book is a double-edged sword:

It is a reference guide for freshers/debutants to guide them for their interview discussions.

It is a reference guide for interview panels for selecting candidates for their practice/units while bringing in standardization in the selection process.

It has more than 600+questions in all the key domains, including a chapter on trending programming languages (Python, R, Ruby and Perl).

It presents an exhaustive question bank with special emphasis on practical scenarios and business cases

This comprehensive title on IT interviews is for aspirants with varying profiles from fresher's to experienced (up to 4 years' of experience) ranging from Engineers, BCA, BSc, B Com and MCA.

It covers all key domains, including Data Structures, OOPs, DBMS, OS, and Methodologies and Processes, Programming Languages and Digital Technologies.

The book covers a section on frameworks and methodology and methodologies, including validations and testing, CMMI, DevOps, Agile, Scrum, APIs, Micro-services and SOA.

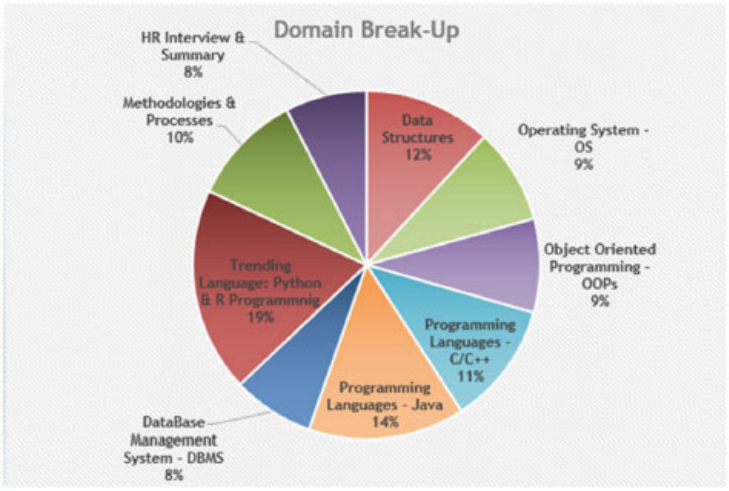
Based on our experience, the assurance is that at least 80% of content will be discussed during a typical interview.

There is a section on pre and post interview preparations.

The book can be selectively read based on the relevant domains.

The coverage is extensive in terms of depth and breadth of a domains addressed in the title.

The title has around 100 diagrams depicting various scenarios, models and methodologies.



Breakup of Q &A's in the book

Chapter 1

Chapter 1 , deals with various aspects and stages of the entire freshers' interview process. It is divided into numerous sections, including strategies and tips for the interviews, freshers' growth path and freshers IT interview process. It describes the qualities that an employer looks for in a potential employee and also helps improve the aspirant's understanding of the interview process. It helps aspirants prepare for the IT interview process and emphasizes on the importance of sufficient preparation.

Chapter 2

Chapter 2 , presents the overview of the written test and the group discussion rounds. It covers the first round, which is the written test and the second round, which is the group discussions stage (or case study or Just a Minute (JAM) round) for the freshers interview process. Apart from explaining the format, it also shares tips and examples for you to practice as preparation for this round. It also covers the objective and goals for the written test and group discussion stages and provides guidelines for adopting best practices and standards.

Chapter 3

Chapter 3 , describes the qualities that an employer looks for in a potential employee and also helps improve the aspirant's understanding of the interview process. It is divided into numerous sections, which include the guidelines to follow before an interview, during the interview and after the interview process. It also has a section on common mistakes done by applicants and the ways to avoid them. It provides strategies and tips for the aspirants for successful interviews. It helps aspirants prepare for the IT interview process and emphasis on the importance of sufficient preparation. It also helps understand the cause of fear and ways to overcome it before the interview.

Chapter 4

Chapter 4 , covers the topic of Data structures in the programming languages, which essentially provides a number of data types and methods that can be performed on these data types. It covers the core concepts, principles, collections, algorithms, programming standards and other key

topics. A data structure is a concrete implementation of the common data types, that is, linked list, stack, queues, maps, and many more, which are covered in this chapter. Different data structures are suited for different application requirements, and a select few are specialized for specific tasks. For example, compilers leverage hash tables to research identifiers where-as RDBMS typically leverage B-tree indexes for data retrieval.

Chapter 5

Chapter 5 , covers the question bank for the operating system domain. It includes core concepts, process management, memory and file management, multi-threading and synchronization.

Chapter 6

Chapter 6 , covers the OOPs aspects and provides the question bank, which includes topics for core concepts, encapsulation, composition and inheritance, and polymorphism.

Chapter 7

Chapter 7 , covers the C and C++ programming languages in detail. The section on C includes a question bank for aspects pertaining to preprocessing, functions, structures, pointer and memory management. The section on C++ includes a question bank for aspects pertaining to core concepts, keywords and operators, constructors and destructors, inheritance and polymorphism.

Chapter 8

Chapter 8 , presents an exhaustive question bank on the Java programming language with special emphasis on practical scenarios and business cases. It covers aspect pertaining to core concepts, keywords and operators, classes, interfaces, construction, destructors, inheritance and polymorphism. It also covers miscellaneous key aspects, including Java iterator, classes, exceptions, JVM, IO, collections and serialization.

Chapter 9

Chapter 9 , covers the DBMS domain. It includes a question bank for various aspects of DBMS. It consists of different sections that cover core concepts, keys and constrains, SQL commands, normalization, stored procedures, functions, triggers, views, joins and miscellaneous DBMS topics that are key & critical.

Chapter 10

Chapter 10 , introduces you to the frequently asked questions on Python and R programming language interviews. It helps you learn all the vital aspects of the trending programming languages. The chapter on Python Programming language covers core concepts, data types andstructures, functions and miscellaneous vital programming aspects. The chapter on R programming covers core concepts, data types andstructures, functions, packages and miscellaneous vital topics on programming aspects. It is a one-stop resource from where you can boost your interview preparations.

Chapter 11

Chapter 11 , presents an exhaustive question bank that covers aspects pertaining to SDLC, verification and validation activities, different methodologies, DevOps, Agile/Scrum, and configuration management.

Chapter 12

Chapter 12 , covers the HR round stage of the freshers IT interview process. It presents an exhaustive question bank with special emphasis on practical scenarios and business cases. It is divided into various sections, which includes the HR round goals and objectives, question bank, do’s and don’ts and business scenarios. It also provides the guidelines to adopt key best practices

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors if any, occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications’ Family.

Table of Contents

Fresher’s interview process (stages)

Written tests

Group discussion, case study round, or Just a Minute (JAM) round

Technical interview

HR round/Managerial interview

Career path for freshers/software engineers

Software engineer/developer

Senior software engineer/senior developer

Lead developer or architect

Project Managers/Sr. Architects – mid-management

Delivery Managers/Principle Architects - senior management

Leadership/Sr. Leadership

Conclusion

1. Written Tests and Group Discussions

Written test

Strategy for written tests

Preparation for written tests

Group discussion stage

Group discussion

Case study

Just a Minute (JAM)

Conclusion

1. Interview Preparations

Guidelines—Before the interview

Guidelines—During the interview

Body language

Guidelines—After an interview

Common interview mistakes and ways to avoid them

Conclusion

1. Data Structures and Algorithms

Concepts

What are generic collections?

Collections

Algorithms – Sorting and Searching

Miscellaneous

Conclusion

1. Operating System

Concepts

Process management

Memory and file management

Multithreading and synchronization

Miscellaneous

Conclusion

1. Object-oriented Programming (OOPs)

Concepts

Encapsulation

Composition and inheritance

Polymorphism

Miscellaneous

Conclusion

1. C and C++ Programming

C programming

Preprocessing

Function and function pointers

Structures and unions

Pointers

Memory management

Miscellaneous

C++ programming

Core concepts

Keywords and operators

Constructors and destructors

Inheritance and polymorphism

Miscellaneous

Conclusion

1. Java Programming

Core concepts

Keywords and operators

Classes, interfaces, constructors, and destructors

Inheritance and polymorphism

Miscellaneous

Conclusion

1. Database Management System

Concepts

Keys and constraints

SQL commands: insert, update, and delete

Miscellaneous

Conclusion

1. Trending Languages– Python and R Programming

Python programming

Core concepts

Data types and data structures

Functions

Miscellaneous topics

Core concepts

Data types and data structures

Functions and packages

Miscellaneous topics

Conclusion

1. Methodologies and Processes

Software development methodologies – Agile and Scrum

DevOps

Software quality

Configuration management

Microservices and SOA

Conclusion

1. HR Round

Goals and objectives

Communication skills

Confidence

Compatibility with the company

Learnability

Question bank

CHAPTER 1

Introduction

With the IT industry changing at a daunting speed in recent years, organizations often look for individuals who are multiskilled, who can adapt quickly, learn quickly, think out-of-the-box and perform duties beyond what is described to them. Therefore, as a fresh graduate, you need to be well prepared to crack a campus interview and make a great start to your corporate career. This chapter will provide you with a high-level view of the campus recruitment process and your career growth path in the IT industry.

The campus interview is the most crucial event in the life of a student. Hence, you must start preparing for it well in advance along with final year studies to stay ahead of the competition.

The main objective of campus interviews is to recognize qualified and competent professionals before they complete their graduation. Campus placements offer employment to the college students who are pursuing graduation or in the final year of graduation. This approach enables the organization to choose the candidates according to their business demands. It is a time-consuming activity, and many companies think it is hard to identify potential talent. Due to projects, assignments given by college and exam schedule, many students do not give the much needed importance to placement training. But it is a fact that students ought to be outfitted in all areas of career advancement along with creating a good impact through the interviews and only that can help them in getting placed in their dream company.

As we all know, preparation is a key to the success and hence, the amount of dedicated efforts one puts in planning and preparations for the campus interview would be critical in successfully cracking the campus interview. It is important to be thoroughly prepared for campus interviews because of the criticality of the interviews in the selection process. It is essential to arrange mock interviews to greatly help cope with tough situations. Mock interviews could be set up together with your network peers. Mock interviews provide time to get ready for the queries that you might receive through the actual interview. This book will help you with interview questions which can be leveraged in your mock interviews so that you are well prepared.

During campus interviews, interview panels focus on interpersonal skills, communication skills, and logical/analytical abilities along with technical competencies. Therefore, it is crucial to not only prepare for technical questions but also be aware of the nuances of group discussions/case study rounds. The HR/Managerial round too is an important step to grab the final offer in hand.

This chapter is divided into numerous sections, including strategies and tips for the interviews, freshers'career path and freshers' IT interview process. This chapter deals with the various aspects and stages of the entire freshers' interview process. This chapter describes the qualities that an employer looks for in a potential employee and helps improve the aspirant's understanding of the interview process. It helps aspirants prepare for the IT interview process and emphasizes on the importance of sufficient preparation.

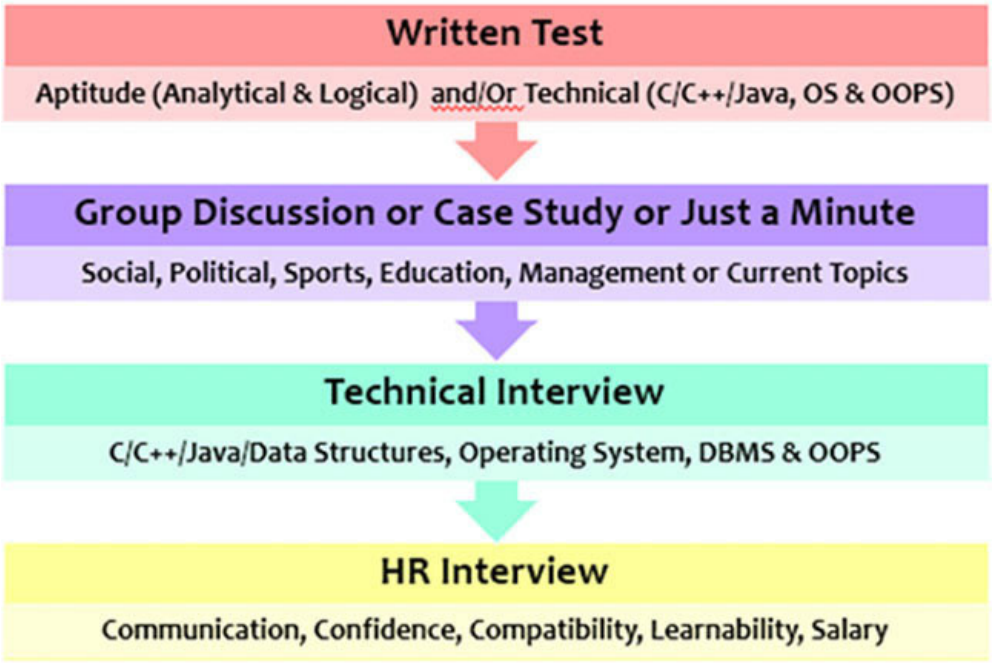
Fresher's interview process (stages)

Now, let us look at the overall process of selection. While different companies have a different format but there are common stages, which are written tests and technical rounds, and hence, it is even more important to prepare for both the written tests and technical rounds.

Before the actual selection process begins, companies conduct pre-placement talks, which are essentially a presentation made by the respective companies covering various topics such as company's vision, structure, services being offered, selection process, package/perks/benefits, and so on. Students should always attend these pre-placement talks to note key details about the company for which they are appearing for an interview. Students can ask questions and get their queries resolved as part of Q & A session, which is usually conducted at the end of pre-placement talks. This will help you prepare well to handle questions during the final interview.

The academic record of the applicant should be good. A minimum of 60% marks in Class 10 and 12, and 65% and above in B. Tech. / B.E / B Sc IT / B.C.A / MCA / M.E / M. Tech, with no backlog is a pre-condition for applying.

Here is how a typical campus recruitment process looks like:



Please refer to the following table, which guides As-On-Date campus process followed in some of the well-known IT companies:

Sr #	Companies	Freshers' interview process – stages				
		Written test	GD	Technical interview	HR interview	Remark – Additional rounds
1	TCS	X		X	X	Managerial round prior to HR
2	Cognizant	X		X	X	
3	Infosys	X		X	X	
4	Wipro	X		X	X	
5	HCL	X		X	X	JAM - Just a Minute Round
6	Tech Mahindra	X		X	X	Story telling round
7	IBM	X		X	X	Business English skills round
8	Capgemini	X	X	X	X	
9	HPE	X	X	X	X	Teach back and managerial interview round
10	Oracle	X		X	X	
11	AtoS	X		X	X	
12	SAP Labs	X		X	X	Managerial interview round
13	Cisco	X		X	X	Written test C, OS, network
14	Persistent	X		X	X	Technical MCQ round, 2 technical rounds
15	Mastek	X		X	X	
16	Mindtree	X		X	X	Coding round
17	Accenture	X		X		
18	Godrej	X		X		2 technical rounds
19	LTI	X		X	X	
20	Mphasis	X	X	X	X	Verbal communication round

Written tests

An aptitude test is one of the initial stages that the applicant must pass to move ahead to the campus interview stage. The aptitude evaluation typically involves assessing logical and analytical reasoning, numerical problems, programming language efficiency, and verbal skills. Speed and accuracy both are crucial to cracking these examinations and this can come only with practice. So, practice and practice more before you appear for this test.

Each applicant may get a different set of questions, but of a similar design and type. The time duration will be typically around 1-2 hours. Different companies have different test patterns and the company representative will share them before the test.

Group discussion, case study round, or Just a Minute (JAM) round

Once you pass the first stage of the written test, the next level of filtering happens in the group discussion round. Again, different companies have a different approach towards the second round. Some organizations give a common topic for group discussion (GD) and some believe in giving IT-related case study for the group of students to come up with solutions/recommendations. The recent trend is also to conduct JAM (Just a Minute) round in which you would have to speak for one minute on any given topic to you. The key objective of this round is to assess communication skills, leadership abilities, listening skills, and teaming aspect of the candidate. Therefore, it is very critical for you to keep yourself abreast on the latest news and current affairs to be able to crack this round.

Technical interview

Post GD, the next important step is to face the technical interview round where a candidate will be assessed for his/her technical ability. Most of the students are rejected at this interview stage. The interview panel will probe you on your area of expertise or competencies. They may cover different programming languages like C++, Java, and Dot Net. For every question, the panel will expect an elaborate response but if you prepare well and go through all the topics/questions covered in this book, there is no reason to worry. You will find most of the technical questions along with answers readily made available for you right here.

HR round/Managerial interview

This is the final round of the process and it focuses on the candidate's confidence and overall personality/soft skills needed for the job. The interviewer may ask questions on education, family background, project experience, strengths, why you should be hired, weaknesses, and so on. The key is to remain poised and handle all questions confidently. Just a small tip here, don't forget to wear your smile.

Career path for freshers/software engineers

The truth is many programming careers have a peak and a decline. Ultimately, it'll get difficult to retain employment as programmers. Many discover this truth flat-footed and unprepared. This section covers important profession information you need to consider making sure that you are prepared beforehand. Research has found out that the shelf-life of a programmer is very brief. The queries these raise are crucial and existential.

What could be the career path of a software engineer/programmer?

What is the future for a programmer/software engineer?

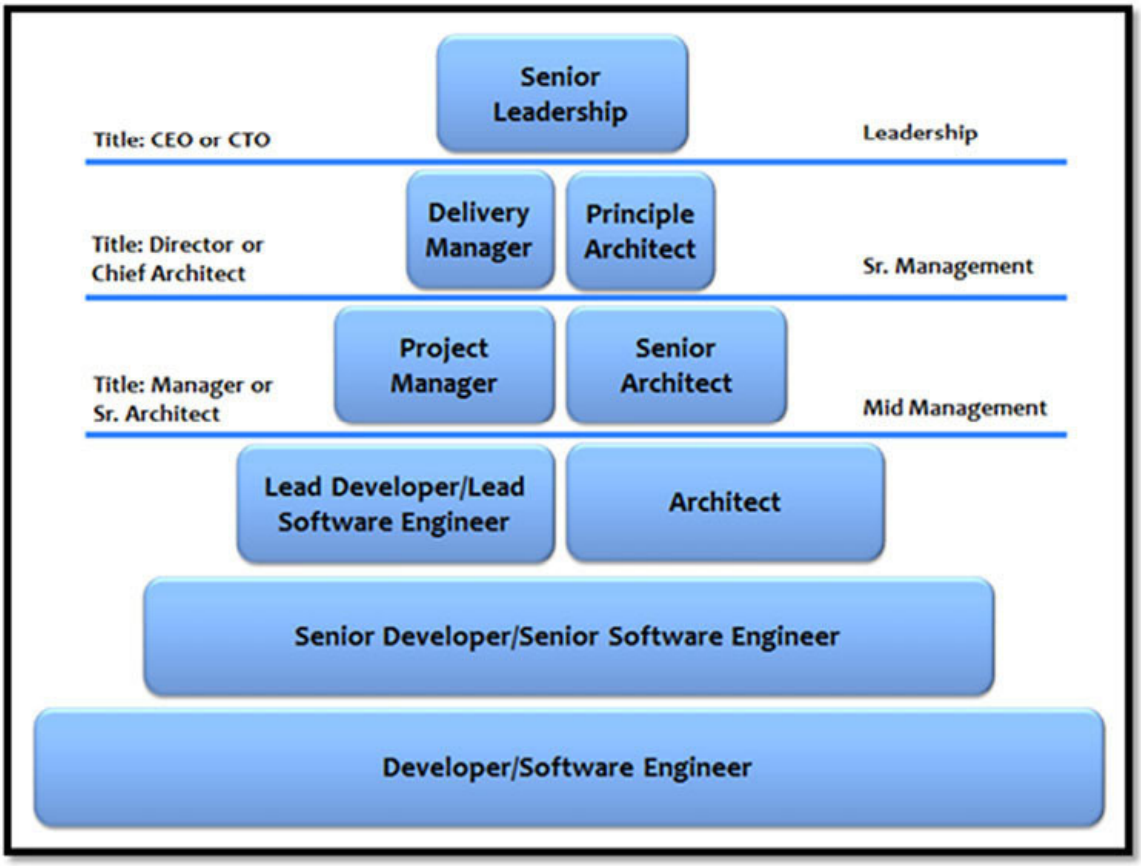
What are career expectations and options?

The universal truth is that eventually, programmers become managers or leaders. Many barely understand the goals and requirements of managerial functions. What does it take to be a manager? And what's the difference between a senior leader and a mid-level manager? What exactly will be the role expectations?

This section shows a potential career path from the start of a career as a software engineer to the apex as a Chief Executive Officer (CEO), Chief Technology Officer (CTO), Chief Information Officer (CIO) .

Many careers stop at different points of the organization ladder and stay there until retirement. Few professions skip rungs of the organization ladder. But leadership and management roles aren't a fit for everybody, and besides one's interest in general management changes over the period. One may hate the very thought of being truly a manager in their 20s, but one might hate the very thought of writing code in their 40s. However, it is good to understand your alternatives and their impacts.

The following diagram shows the growth path for software engineers showing both technical and managerial roles:



Software engineer/developer

The following competencies are required for this role

0-4 years of experience

Preliminary understanding of the whole SDLC

Experience in C/C++, Java, or Dot Net frameworks

Preliminary knowledge of application services and databases

You should make most of your time to learn and deep dive into technology as a software developer. It is also important to gain industry understanding while you learn technology. This is the crucial time, one needs to build a strong foundation by learning the right coding practices, software style patterns, design patterns, and so on.

Senior software engineer/senior developer

The following competencies are required for this role

4-10+ years of IT experience

Deep understanding of a whole SDLC

Advanced knowledge in C/C++, Java, or Dot Net frameworks

Deep knowledge of application services and database

Comfortable working in any application domain

Senior software engineers/senior programmers are proficient at the entire software stack development. A lot of a programmer's profession is normally spent as a senior programmer. If you dislike management functions and like to code, you may be considered a senior programmer for the whole career. This is the function that gets harder to compete as we get older.

This role may also be a jump-off point for another rung on the ladder. Once you gain technical and domain expertise, futuristically you may plan

your career towards becoming an architect, a chief technology officer or a founder of a startup. So, a deep understanding of the technology stack will go a long way in achieving this career goal.

Lead developer or architect

The following competencies are required for this role:

7-10+ years of IT experience

Similar expertise as a senior developer

Lead developer: Transitionary role into an architect or mid-level manager

Architect: Non-transitionary technical role

After 10+ years of programming expertise, if the management isn't your thing, an architect may be the highest rank on the organizations' technical ladder. Architects do write programs, but more frequently, they design complicated with complex applications that will be implemented by groups of senior and junior programmers/developers. Architects normally leverage the wisdom they have gained after years of experience to create successful software applications and systems. As new business requirements come in, the architect should be conversant with the methodologies and frameworks to build and scale applications and systems in a variety of domains.

A lead developer is predominately a senior programmer that additional junior and senior programmers look up to for direction and guidance. Though the lead developers do not generally hire or fire programmers, they engage in a lot of very similar work as managers. They orchestrate all the tasks and are the decision-makers for implementation while also composing code.

Project Managers/Sr. Architects – mid-management

The following competencies are required for this role

Titles include Project or Product Manager or Sr Architects

7-10+ years of IT experience

Communication, Leadership, Team management, Negotiation

Deep understanding in both business and IT

Reports to a senior management

Management may traditionally be the next rung of the ladder for software engineers. There are different focus areas for management. In case you are passionate about features and product enhancements, then being truly a product manager is a solid fit. In case you are detail-oriented and like managing and monitoring workstreams, a project manager function is a right fit.

The normal role of the project manager is normally to mediate the requirements of the delivery team and product manager. This role needs important people management skills, skills at mediating conflicts, performing and delivering as a group.

Delivery Managers/Principle Architects - senior management

The following competencies are required for this role

Titles consist of Delivery Managers or Principle Architects

10-15+ years of IT experience

Reports to Leadership/Sr. Leadership

Deep understanding in both business and IT

Leadership and negotiation skills

As one goes higher up the ladder, the much less you will be a hands-on SME. At the top, it is majorly about people and program management. Mid-level managers still get to dip their toes in technology, but the senior management spend most of their time strategizing, leading, inspiring, and motivating the team around.

Principle Architects are accountable for driving the CIO's vision and strategy. This deals with defining strategic programs, selecting technology, and defining roadmaps. These architects help the CIO to make sure the IT investments for the organization are aligned with the business goals

and objectives, thus providing a competitive advantage for the business. They are also accountable for defining the rules, standards, guidelines, and investing in a governance methodology to align the implementation of the described guidelines and requirements. In a few companies, this role could be merged with the CIO and may have the title - chief architect. That is true for most platform and product businesses.

Leadership/Sr. Leadership

The following competencies are required for this role

Title consists of CTO, CEO, or VP

15-20+ years of IT experience

Reports to Senior Leadership or the Board of Directors

Experience in governance

Leadership and negotiation skills

The key difference between a senior leader and a mid-level manager is that senior leaders lead the mid-level managers. The managers won't need to be managed and are often required to be led. Mid-level managers should get high-level guidance from the senior leaders that they have to drive the unit towards a common goal, rather than get turn-by-turn directions.

Senior leaders make high-level decisions and inspire their teams to follow those decisions and have confidence in the mission. Senior leaders spend all their time on larger things: strategizing, inspiring, leading, and motivating the team around.

The leadership/senior leadership makes certain that everyone in the organization is rowing in the same direction, ensuring it leads to the shore and making sure people are aware of why they are rowing in that direction. That is an extremely difficult task and is a rare skill to possess.

Conclusion

This chapter covered the introduction, strategies and tips, growth path and the interview process in detail.

The next chapter covers aspects related to the written test and the group discussion rounds for the freshers' interview. This also covers the objective and goals for these stages and provides guidelines for adopting best practices and standards.

The opportunities for software engineers/programmers today are plentiful and rich. Programmers will be in higher demand; in fact, it is a great era to be a programmer. So good luck to you all!

CHAPTER 2

Written Tests and Group Discussions

This chapter initially presents the overview of the written test and the group discussion rounds. This chapter then covers the first round which is the written test and second round which is the group discussions stage (or case study or Just a Minute (JAM) round) for the freshers’interview process. Apart from explaining the format, this chapter also shares tips and examples for you to practice as preparation for this round. This chapter also covers the objectives and goals for written tests and group discussion stages and provides guidelines for adopting best practices and standards.

Written test

An aptitude test is an assessment that evaluates the ability and potential to execute a certain task. This test has a standard method of scoring where results are quantified and compared.

In the first stage of the campus recruitment process, many companies conduct written tests to be able to identify the competencies of the candidates. These tools are leveraged by organizations to exclude applicants that usually do not match the company requirements. As we’ve seen, the number of aspirants is usually higher than the number of openings in organizations, and this leads to companies using such methods to eliminate candidates. Generally, companies may conduct any or all of these kinds of written tests, which are as follows:

Aptitude tests: Most companies assess the candidate on analytical and decision-making skills. Vocabulary and language skills are also checked in the written test to measure the candidate’s communication skills. Basic mathematical skills, which focus on applying concepts in relatively complex situations, are tested to judge the analytical skills.

Technical tests: These tests include technical questions in a particular domain of a student’s specialization such as C++/Java/Dot Net/Databases. These tests measure the technical skills and subject matter knowledge. Individuals with strong fundamentals of these subjects perform well in these tests.

Strategy for written tests

The overall structure of the written tests is such that candidates get significantly less than one minute to answer a question. The expectation is usually that the applicant will maximize his rating by selecting questions judiciously. Companies may structure the test into different sections, which might consist of quantitative, verbal, data interpretation and logical reasoning. Companies adopt two methodologies to conduct the technical assessment: it either is part of the aptitude test or conduct it as a separate test validating the domains of importance. Organizations generally do not adopt the criterion of selection-based cut-offs. Keeping this in mind, one should understand that the score in the examination is the most crucial one. The guidelines are usually to validate your strengths and weaknesses and make an effort to attempt questions; one is most comfortable with. You can invest time on questions where you are assured of scoring the maximum marks in the least amount of time.

Preparation for written tests

The written test round checks a potential applicant on several facets such as logical and analytical ability, subject matter expertise, communication skills, and more. To successfully clear this round, you should accomplish the following:

Aptitude building: Ensure that you improve your aptitude by leveraging the very best preparation materials. Aptitude building involves dealing with hard-core topics of Mathematics, English, and the basics. Systematic research and study in the long term is the right methodology of enhancing one's aptitude levels, including focusing on the fundamentals of these areas.

Focus on concepts and fundamentals: Focus on domain fundamentals and concepts, and you are completely ready to challenge this section of the entrance exam.

After successful completion of the first round of the campus recruitment process, the next round consists of the group discussions/casestudy/JAM and technical interview . The details are explained in the next section.

Group discussion stage

This section describes the process around group discussions, case study, and just in minute rounds.

Group discussion

Group discussion (GD) is one of the most effective tools to choose potential applicants from a comparative perspective. The GD methodology is usually leveraged to gauge if the applicant has certain personality characteristics and abilities that are preferred in its potential workers. In the GD process, several potential applicants are given a subject or topic and given a couple of minutes to elucidate the storyline, and then asked to discuss the topic in a group for about 20-25 minutes. Subject matter experts (SMEs) do the GD evaluations for the candidates based on these group discussions. The subject matter expert prepares a report after analyzing and consolidating the facts by the end of the group discussion.

The key reason why group discussions and interviews are conducted, after testing the technical and analytical skills, is to understand you as an individual and gauge how you would easily fit into their organization DNA. The GD evaluates one's style to function as a unit or in a group. As an associate of an organization, one is expected to function in the team environment. As a result, how you integrate and coordinate within a team becomes an essential criterion of the selection procedure. Managers need to lead a unit and get the best outcomes out of all the associates. That is the reason why the management institutes include GD as part of the selection procedure.

In a GD, a subject is given to the peer group for discussion. The topic could be controversial, factual, political, abstract, and many more. Controversial topics are argumentative and they measure the patience of the applicant. Practical topics are based on facts that provide a candidate an opportunity to prove that he's aware of specifics. These feature topics like tourism, education policy in India, and so on. Occasionally, GDs can feature abstract topics. Abstract topics check a candidate's lateral thinking and creativity. In the GD rounds, potential candidates are evaluated on the following parameters:

Capability and effectiveness in presenting your views on the topic

The depth of the knowledge and trends regarding the subject

Leadership, communication, and coordination abilities

Originality and effectiveness one brings to the discussion

Ability to address the whole group

Ability to articulate/summarize the topic at the end

Listening skills/active participation

Behavioral aspects/versatility/managing conflict situation

Leading the conversation/focus on the topic

All these above points look pretty simple; however, the truth is in the heat of a discussion, potential candidates at times forget these basic points and end up taking criticism personally. It is critical to remember that this is a group discussion and different faces of the issue must be brainstormed before arriving at any conclusion.

One key thing to understand is the rules that apply to informal discussions also apply in formal GDs. A solid knowledge of current affairs and happenings will help you excel in the group discussion stage.

List of potential GD topics

Here is a list of GD topics that are generally used for the interview:

Current topics:

Will digital payments ensure the Indian economy to go cashless?

Can India dream of hosting the Olympics?

Do we really need Smart Cities?

Is the youth of India confused or confident?

PM's vision - India a Manufacturing Hub - dream vs practical possibility

Youth in politics

Economics and business:

Are Indians less quality conscious?

Commercialization of health care: bad or good. Discuss.

Are MNCs superior to Indian companies?

Privatization will lead to less corruption. Elaborate.

Discuss globalization vs nationalism.

General interest:

Braindrain has to be stopped.

Flexi timings or fixed timings?

Science is aboon or bane.

How to deal with international terrorism?

Is dependence on computers a good thing?

Management topics:

We need more entrepreneurs than managers.

Indians perform better as individual contributors than in teams.

Positive attitude and not knowledge is required for business success.

Whether smartworking or hardworking is desirable?

Women are better at multi-tasking.

Do women make good managers?

Creative topics:

Ship docked in the harbor cannot face storms.

Do brains and beauty go together?

Every cloud has a silver lining.

Rules are meant to be broken.

Social topics:

Terrorism in India.

Religion should not be mixed with politics.

Should we ban smoking completely?

Influence of social networks on youth.

If winning isn't everything, why do they maintain the score?

Is T20 cricket killing real cricketing skills?

Case study

Some companies conduct the case study round as an alternative to GD. It is typically included in a scenario where an employee is expected to work directly with the end client, so most of the consulting firms include this round as part of their selection process.

The casestudy round is conducted in a single session where candidates are divided into small groups and are given a real-life business scenario/issue, which is typically handled by the company with its client. Candidates are expected to understand, analyze the business issue and provide recommendations or solutions to the stated business problem. Let us look at the following case study formats:

Offline discussion: This is a discussion when groups discuss about the issue and potential solution among themselves. Individuals are observed silently by the interviewers during the offline discussion and they are assessed on key aspects such as listening skills, assertiveness, taking initiatives, and teaming abilities.

Presentation: In the second half, they collectively present their solution using the PowerPoint presentation as it is done in a real-life client situation. It is expected that everyone in the group shares/presents his/her perspective on the solution. At the end of the presentation, question and answers round is conducted. During the presentation round, they are evaluated for their presentation skills, communication, and articulation for their ability to handle difficult client conversations.

List of potential case study topics

Build a business case on how you can solve a given business issue by building a new product, new service or solution.

Develop a business strategy in a scenario where the client is going in for merger and acquisition.

Develop a business strategy on how you can help the company in its business growth, performance improvement, and many more.

Develop a strategy for business transformation/digital transformation.

This case study round helps to identify the best among a pool of best candidates and hence, it is the most preferred approach by many leading multinational firms.

Just a Minute (JAM)

Just a Minute (JAM) round is a new alternative to the GD or case study round wherein every individual gets to speak for a minute on a given topic. JAM helps in assessing a candidate's skills such as his/her presence of mind, communication, time management, brevity, and articulation. As part of the JAM process, students are divided into groups and a JAM master is assigned. The JAM master provides one topic to each student and they are given 30 to 45 seconds to gather their thoughts and prepare on the topic. One minute is allocated to each student to speak on the topic.

Best practices to successfully clear the JAM round:

First and foremost, keep yourself updated on current affairs, news, and improve your general knowledge.

Gather and structure your thoughts, topics, and key points prior (during preparation time, that is, 30 to 45 seconds).

Make eye contact, maintain a positive outlook, smile and be confident.

Avoid long, complex sentences when you present your point of view; instead, speak in simple English with conviction and with clarity of thoughts.

List of potential JAM topics

Do we need net neutrality?

Digital disruption – A blessing or a curse?

Is data on the cloud safe?

Can AI replace teachers?

Role of women in economic growth.

Being workaholic is toxic.

Conclusion

The skills that are needed for the group discussion stage are the typical employability skills that help you to stay in a job and work your way to the top. While there will always be some job-specific skills that an employer is looking for, most employers will also want you to have these skills.

These skills are not taught as a special curriculum in the academic curriculum. They need to be acquired. These days colleges and a few soft skills institutions provide coaching on these skills.

Acquiring such skills is an ongoing process and if one starts early, he/she will have an edge over others. If the students start identifying where they stand in these and based on this self-knowledge, start building upon it, no matter what stream they are in, they will become employable.

In this chapter, we covered aspects of the written tests and group discussions.

In the next chapter, we will cover aspects related to the interview preparation process in a step-wise manner, that is, the guidelines to follow before, during, and after an interview.

CHAPTER 3

Interview Preparations

Once you are through with the written test and second round of the GD/case study/JAM, the next round is the technical interview and final round of HR/Managerial interview. This chapter will help you prepare for these interview rounds. Apart from providing the tricks and techniques for an effective interview, it also provides guidance on how to prepare for the campus interviews, analyze the organization and job profile, and helps you understand what makes you the ideal match for this position. Read these tips to prepare you prior to these competitive interviews.

If you look at it, the company has so many applicants to select from who have virtually identical degrees of technical competencies, so how do you differentiate yourself from the crowd in such a situation; the only way out is to stand out with depth of your knowledge and with good articulation or verbal skills. This is also the reason why technical and the HR/Managerial rounds become differentiators to shortlist the potential applicants.

Consider this analogy. If an organization was a machine, the specialized competencies become the core machinery and the soft skills would be the lubrication and therefore for a smooth ride, we are in need of both. Therefore, both technical and final HR/Managerial rounds become critical in the fresher's campus recruitment process.

This chapter is divided into numerous sections which include the guidelines to follow before interview, during interview and after the interview process. This chapter also has a section on the common mistakes done by applicants and the ways to avoid them. This chapter describes the qualities that an employer looks for in a potential employee and also helps improve the aspirant's understanding of the interview process. It provides strategies and tips for the aspirants for successful interviews. It helps aspirants prepare for the IT interview process and emphasizes on the importance of sufficient preparation. This chapter also helps understand the causes of fear and ways to overcome them before the interview.

Guidelines—Before the interview

There is one complete solution to overcome the interview fear which is preparation, preparation, and only preparation. Preparing prior to the interview will increase your confidence. The crucial elements that you need to know prior to the interview are as follows:

Skillsets you are proficient about

Domain/Platform you want to select

Strengths and weaknesses

Here are some techniques and tricks for freshers aspiring to join the community of working professionals and to give that first very best impression through the interview.

Eat and sleep well

To avoid feeling starving and sleepy, one needs to sleep and eat well prior to the interview day; otherwise, throughout the interview, your lack of sound sleep may convert to headaches, stress, and tiredness as well as your hungriness may convert to bellyache. Such complications divert your mind from the actual goals and may spoil your interview discussion.

You have to sleep for at least 7-8 hours at night to keep the mind healthy and fresh. The greatest thing to accomplish a day prior to the interview is eat healthy meals, sleep early at night, and get up early the next morning. An early morning fresh air makes the mind healthier and in addition, it helps you recall all that you have planned and prepared for the interview.

Dress appropriately

Please remember that you would meet the interview panel in the interview for the first and you need to leave your first impression on them. In case you are well dressed, you can make a good first impression and the panel will feel like speaking with you more.

For the interview, wear suitable or ideally light color attire that suits your overall personality. Make sure that your shoes are polished, hair is well combed, and nails are trimmed and remember to always carry a handkerchief and pen.

The kind of outfit you wear speaks about your personality like clothes sense, etiquettes, features, and many more. Also, the kind of attire you put on says a whole lot about your professional etiquette and the interviewer's initial judgment begins from there itself. Regardless of, if the interview is on the phone, or if it is really is a one-on-one interview, dress up appropriately for just about any interview because while giving the phone interview, the interviewer might just request you to come on Skype or any video messenger.

Make sure your dress is usually neat, tidy and properly pressed. The entire outfit should match with one another. It should not be ripped or there should be no holes in the outfit. The ideal attire for men is a two-piece suit and tie and for women, it is a skirt and shirt or trousers and top. If you're confused about what to wear for the interview, get in touch with the employees of an organization and inquire about the attire required for the interview.

In addition to the dress, the shoes should be polished, the hair ought to be properly trimmed and groomed, and the hands and face should appear neat. Wear minimal jewels. Men can wear a watch, which is adequate, and women can wear a watch, earring, and do some light makeup, which is enough to give a professional look.

Curriculum vitae and certificates

Curriculum Vitae (CV) is an extremely critical part of the interview that reveals how you were in academics and other competencies. Make sure that you carry multiple copies of your CV. It must be tuned as per the job requirement. You need to know every detail mentioned in the CV. Also, the qualification and experience ought to be well-structured and very easily understandable. Customize your CV by understanding the work requirements.

Remember, together with the CV you need to carry the originals of your educational certificates. At the interview, you may also bring your appreciation certificates. These appreciation certificates may boost the percentage of getting your dream job.

Prepare to ask relevant questions

When given an opportunity by the panel, ensure that you ask intelligent questions. This would require you to research on the job requirements and the organization very well prior to going for the interview. Not asking questions in the interview demonstrates your insufficient interest in the position or being dull.

Prepare for the behavioral questions

Behavioral questions are meant to know your experience and competencies. These questions intend to put forth your real-self. For example, tell us how you handled the examination pressure, or how did you manage to complete the project while attending college.

Online persona

Employers leverage the internet as a tool for searching the right candidates. Ensure your online profile is up-to-date and includes the required keywords to catch the attention of the potential employers or headhunters. Do not forget to update your information.

The LinkedIn profile has to be filled out and synced with your resume. Employers use the LinkedIn profiles for checking the applicants. Highlight

the job requirements and your unique talents. Understand that your online profile represents you in the real world. Configure Google alerts or other alerts to receive notifications if you do not want to miss any new opportunities to influence the potential employer and do not forget to add recommendations.

Social media provides you a great opportunity to build networks. It is necessary to get visibility on various social media networks if you want to be found by people who share the same niche as you.

Communities like Quora, Stack Overflow, and many more enable you to create an online persona. Join these communities and be active. It will not only help you interact with experts but will also improve your problem-solving skills. Sites like Stack Overflow and GitHub allow users to work together on projects that can open many opportunities to learn.

Another great way to nurture a personal brand is to contribute to open-source projects. Contributing to open-source projects can be an amazing experience but it will help earn you recognition among the fellow community.

The key forums and communities where one should participate are as follows:

Stack Overflow

GitHub

Quora

Medium

CodeRanch

StackExchange

Correlate your skills with the job requirements

Read the job description thoroughly to learn what the company wants in an applicant and if the job description fits your own professional and personal qualities. Once you perform your analysis and look for an ideal match, make a summary of abilities, personal and professional qualities required by the job.

The correlation between your skills with skills necessary for the opportunity maximizes your likelihood of influencing the panel and goes nearer to the offer.

Research the job description and company

Prior to the interview, spend quality time to investigate information pertaining to the organization, its services and products, and the job description. Doing research and study, you would be able to collect the entire company’s details pre-dominately from the company website and in addition from other resources like Glassdoor and many more. Additionally, you will see many other assets to articulate genuine information regarding the company and the role. You just need to find the correct sources to begin with. The more you study and research about the organization and the profile requirement, the better you'll be in answering the interview panel questions.

This is actually the critical section of the interview where the interviewer panel will like to know about your interest about the company and its culture. One's understanding of the organization and its own products and services will take you nearer to the offer.

This knowledge about the organization also helps you in understanding if the organization and its own culture certainly are a good fit for you personally. Hence, before attending an interview, spend some amount of time in discovering and gathering maximum data about the organization.

One advantage of a campus interview process is that the companies introduce themselves just before the interview process starts as part of pre-placement talks. Therefore, please make it a point to sit through these pre-placement talks and listen attentively.

Practice ahead before the interview

Practice makes us perfect and hence, sincere efforts and practice will not go in vain, and shortly will open the doors to success. You might or might not crack the test in the first attempt, but never quit. Make an effort to correct the errors you made in the first attempt and move on.

Similarly, if the employment seeker sets an objective for his dream job, he/she must keep practicing if the interview is scheduled or not scheduled. You won’t know when one might receive a call for an interview and later, you cannot complain about the brief span of time to prepare and appear for the interview. Many applicants find it hard to prepare in a short span and constant practice relaxes your nervousness and increases your confidence levels, which may be easily spotted by the interview panel.

Before an interview, as you prepare for other activities, spend ample time in practicing interview questions and answers through this interview

guide. This planning and preparation will help you in framing your responses and overcoming the nervousness through the interview.

If you're shy and more likely to face difficulties throughout a face-to-face round, start practicing prior to the interview by talking to family and friends or standing before a mirror.

Know thyself - strengths and weaknesses

Not many people are good at all domains. Every one of us has our abilities, strengths, and weaknesses as well. However, that's not an obstacle. Companies look for individuals who know their main area of specialization and are subject matter experts. As a result, it pays to become an expert in a few areas if not really the jack of most.

The common mistakes many make are boast about of knowing a domain which they have not had any knowledge. Remember heavy resumes are as difficult to articulate as they are to produce. So, determine your competencies and maintain your resumes simple and concise while also understanding your limits.

Before coming to the interview, articulate in advance your core competencies, strengths and weaknesses. In the interview discussion, tell your strengths by giving examples, but don’t pause in between. Otherwise, the interviewer might think, you are cooking it up. Do emphasize on your strengths that are relevant to the job profile.

Prepare well for the HR/Managerial interview

More than 50% of applicants flunk the HR/Managerial interviews. When you have performed well through the technical round and you think the job is yours? This is an incorrect assumption. You still have got the HR round, and a lot more than 50% of applicants flunk the HR stage. The HR round is conducted to comprehend the behavioral aspects of the applicant and find if he/she would fit into the organizational culture. Make sure that you have comprehended the real reason for each question and take a correct approach to respond to it.

Prepare to answer common questions in advance

Imagine that two minutes into the interview conversation and you are still adjusting to the interview panel’s glances when you are asked, tell me something about yourself. This sounds fairly simple, but this query is among the hardest to respond. You would not want to stammer at this stage, and at the same time, you usually do not want to parrot your CV back again to the interview panel.

Questions like why do you wish to work for our company? Or why have you got this much CG? Or where do you see yourself 5 year from now? These are common questions, and they could help you if you plan these beforehand. To avoid last-minute meltdowns, plan such common questions beforehand.

Don’t whine about the unfair practices in your interview conversation. The interviewer isn't your shrink. He's just trying to look for the perfect applicant for the job opportunity. Long story short, simply focus on your abilities and see how you can add value to the organization.

Interview etiquettes

The easiest method to learn etiquettes prior to the interview is by meeting and speaking politely with qualified professionals, family, and friends who know the etiquettes required for interviewing. While talking to them, you can inquire further for the opinions about behavior and body gestures.

While finding your way through the interview, you can even stand before a mirror to know your self-confidence level and body gestures. Throughout the interview, shake the hands firmly, maintain eye contact with the interviewer when you are articulating the points, sit straight, be relaxed and give considerations.

Avoid reaching late for the interview; ensure you have researched the right path to the interview location, the time it would take to reach there, and choose the best route there. Take a few extra minutes and reach the interview destination early. Nothing at all frustrates the interviewer more than a candidate appearing late.

In the event that the interview location is remote, reach the location a day in advance, so that, on the day of the interview, you can reach the destination on time.

Guidelines—During the interview

As much as it is important to prepare in advance, it is also very important how you speak and conduct yourself during the interview. Creating your first good impression is very important; this sets the precedence for the rest of the interview and dealing with a mistake made here is extremely difficult and you may have to spend the rest of the interview trying to regain your lost ground. Here are some tips for you to follow during the interview.

Body language

When you meet up with the panel for the very first time, don’t forget to give them a firm handshake and smile. While shaking hands, the panel should believe that you are ready and assured about the interview.

The most crucial things that need to be looked at before shaking hands are as follows:

Give a firm handshake but make sure that you don't hurt them.

Once you hold the hand, shake it a few times.

You ought to be the someone to extend the hand first.

Ensure that the hands are clean and wiped. Otherwise, the panel may believe you lack manners.

Keep a tissuepaper/handkerchief in your pocket.

Make eye contact with the interview panel through the handshake.

Meet your interviewer with a pleasant smile and a firm handshake. Mostly, the body language and eye contact present the level of confidence in you before the interview. This age old formula showing confidence still works wonders. Eye contact is an effective way to allow the panel to understand that you are on the same page

Try to avoid following gestures such as moving fingers on hair, moving the body while sitting on the chair, biting pens, pencil, nails or other objects, shaking legs, touching face by hands, hiding face by hands, and many more.

Stay calm and confident

Clear your diary before and after the interview, so you can stay fully focused on the event. In order to give your total attention on the interview, you don’t want to be preoccupied with other things that need to be done on that day.

Naturally, nerves will begin to kick in, but try to stay calm and confident. Take deep breaths and practice positive, self-affirmation thoughts in your mind. If you’ve done adequate research, you shouldn’t have anything to fret about.

First impressions count

It’s a fact that first impressions count, so as soon as the interviewer approaches you, ensure you come across as personable, professional and courteous. Stand up straight, smile, make eye contact and extend your hand. Say hello and introduce yourself stating that it’s a pleasure to meet them, and thank them for taking the time to see you today. Attention to detail can go a long way to securing your dream role.

Strong listening skills

In the case of a job interview, strong listening skills are just as important as other skills. How well one pays attention during the interview, will pave the way to optimum with positive responses.

During the interview discussion, pay attention to your responses. First, pay attention, listen carefully and clearly to what interviewers/panels are asking, and then compose an appropriate response. Your responses should convey that you are highly interested and believe that this job is an excellent fit for your skills and competencies.

Listening is normally an integral aspect of communication. Nonetheless, it will be good if you have an impressive oration style and a killer vocabulary; however, the conversation is usually incomplete without listening. During an interview, make sure to respond only after having comprehended the question(s) correctly.

Stay calm and cool

Whether you are giving your first interview or have already given many interviews, during the interview, try and handle all the questions calmly even if you know the response to the question or not. This specific characteristic will not only help you handle tough questions, but it will also create a cool and calm image in front of the interviewer panel.

During the discussion, sit upright and pay attention to the questions very carefully and just answer the questions after taking a pause and consolidate your thought process. Answer the question if you are sure about the reply and have understood it correctly, or say sorry, I really do not know the answer. Don’t be overconfident. If you're confused about certain factors, don’t display your indecisiveness using filler phrases and it is best to borrow some more time from panel to take into account the right response or ask for clarifications.

Exhibit communication skills

One of the most important qualities interviewers look for in potential candidates is the ability to communicate, that is, how good is the candidate at orating and articulating. Software development is a teamwork wherein you are required to interact regularly with your peers and stakeholders from different organizations. So, it is extremely critical that you know the language well. Ensure that you work on your English by regularly reading the newspapers and watching news channels like CNN and BBC. Ensure that you are well prepared to take a grammar test or vocabulary test if the need arises during the interview.

Be confident and stay humble

Show enthusiasm and confidence during the interview discussions since you will be required to work in a team. It is extremely critical to be pleasant and humble. This is one of the qualities recruiters would want to see in a potential candidate.

Be enthusiastic and energetic once you are in the interview room, and don't allow the panel to discover your anxieties and nervousness if any. Remember you'll be observed, so be calm and confident. Don't lose your heart and courage soon, even if the interview is normally taking longer and there is no job offer yet rolled out. In the event, if you start doubting your potential, your nervousness could overpower your characteristics, and you'll lose even though you are eligible.

Calmness displays emotional maturity about the candidate. Remaining relaxed in a job interview is a hard proposition, but then that's what's required. Calmness will not imply becoming apathetic or unenthusiastic in the interview but realizing that you are anxious and not letting it come in the way.

Applicants should also watch out for the impressions made in nonverbal conversations. Facial expressions and your body language can assist you in establishing an excellent rapport with the panel. Gestures pause and silences may all show everything you understand, mean, or wish to emphasize.

For just about any job positions, self-confidence becomes an integral factor in the initial discussion of the job interview because employers usually do not want to recruit a shy applicant who is doubtful by nature for a particular job position.

Be positive and never lie

Avoid speaking negatively about any of your experiences or interactions. Ensure that you display enthusiasm for the position you are being interviewed and that you are keen to work in the organization.

With so many social media sites and the organizations having access to personal investigators, it is simple for the company to verify things you said during the interview. So, make sure you do not falsify the details about your candidature at any point in time. One could easily get the job by telling lies; however, the day it is exposed, the only door open would be the exit door.

Don't bluff in your technical rounds too. In case you are confident, then answer otherwise, mention politely, for example, I am not exactly sure of the response but let me tell you what I understand about the domain and make an effort to make an informed guess.

Remember that knowledge is usually immense, and you are bound to have some restrictions. The interview panel has the experience to believe otherwise. So, simply give your very best to the opportunities that are offered to you. If you're honest about your limitations, there are high chances that you would be appreciated.

Take advantage of your credentials

Usually, the interviewer will ask questions from the information mentioned in your CV such as educational background, work history, projects, etc. Thoroughly verify the details mentioned in your CV. There should not be any mismatch between the information you provide during the discussion and the information that is mentioned in your CV. In case the interviewer spots a variance, he may conclude that the candidate is lying, and his/her profile is not genuine.

A candidate gets extra credit when he/she has taken a degree or has done certification on the skills required for the job or taken any special training. For example, if an organization needs a software engineer in C, C++, Java/J2EE, and SQL. A candidate who has done his/her engineering from the Computer Science stream and has a certification in Java or .NET will be an ideal candidate for the IT organization.

Take advantage of your educational degree, certification and training courses, if you have done any. Don't forget to emphasize on such credentials in the interview. Your credentials will increase your confidence level and help you in answering the questions effectively.

While answering, try to accommodate appropriate case studies with your responses. While giving out the case studies, try to relate it to your experience and say how you handled those scenarios effectively. The more skilled you are there are more chances of being considered for the job role.

Your responses should be to the point and direct. Show your flexibility in accepting all challenges that you may face as an employee of the organization. Your answer should present your skills and ability and how you are going to leverage those abilities and skills in providing value to

the organization.

Handling awkward questions

The panels may ask the applicant the domain they wish to be interviewed upon. Eureka!! Here is a golden opportunity. Under no circumstances, choose a challenging domain or subject you know very little about; rather select the domain that you will be most confident of.

Don’t get defensive if the panel asks you something you find challenging. Handle such circumstances very intelligently through the discussion. Generally, the interviewer is judging your reaction to stressful circumstances and tolerance levels. That is the kind of tension interviews are made from.

You never know if the panel is planning for a stress interview. If the panel shows up to be rude, keep your cool and don't lose focus. If you're unable to answer a few of these questions and don't panic under any circumstances. Occasionally, the panel will ask hard questions. It is OK. Just to share an experience, after I had answered one complex question during the interview correctly, the interviewer brought up some new research being carried out in a field and mocked me for being unsure of enough. Occasionally, they ask questions, which are probably too much for a graduate college student. It is simply to know in case you are well prepared and have a curiosity beyond what's taught at your university.

Show interest

All through the interview, show a keen interest in working for the organization; even if you are unaware about the job profile or you don't know whether you are keen to get that job. Acting and showing curiosity can engage the interviewers for additional time that will help you to understand about the organization, interviewers and the job profile. Also, it can benefit you to consider the correct decision about the job that has been offered to you.

Ask intelligent and smart questions

By the end of the interview discussion, applicants get a chance to ask a few questions to the panel. This chance is given to the applicants to check on their keenness towards the job profile and organization. Therefore, without missing this golden opportunity, ask intelligent questions that can show curiosity in the organization and job profile. But don’t switch the question towards the compensation and organization benefits; otherwise, the interviewer may believe that you only need the job for an excellent pay.

You can ask the questions like if I get employed, exactly what will be the challenges I will need to face to be a top performer? And how does the company motivate their work force and employers to understand the company value? How can I help in achieving the company goal?

Listen thoroughly to the campus presentations held prior to the actual interview process starts. If you are provided the opportunity to ask questions, usually do not hesitate and show just how much you are already acquainted with the organization as well as your job profile. The questions you can ask will be, for example, about the interviewer’s profile, about a typical day at work, what qualities they search for in a prospective employee or something exclusive you noticed through the presentation.

Guidelines—After an interview

While waiting for a response from the hiring manager can definitely be frustrating, knowing what to do after an interview can actually help you influence the outcome. Once you’ve accomplished these steps, you can be rest assured that you did everything in your power to get the job offer.

Drop a thank note

Always, thank your interviewer after your interview so do drop a thank you note. The more contact you have, the more you remain in their minds over the other applicants. It shows you’re thoughtful, motivated, and that you care.

Create a checklist

Create a list of items that you did well and that you would like to improve on. This can be one of the most effective things to do after an interview because it will allow you to improve for future job interviews, or assess general weaknesses that will be helpful in your career. Doing this right after an interview is ideal as well because everything will be fresh. It doesn’t have to be a big fancy list, bullet points will do. The most important thing to remember is to be brutally honest with yourself. Even if you didn’t like the way the interview was conducted, there’s always something you can do to improve. Find it!

Follow up appropriately

Sometimes, the interviewer will tell you when they will notify you; other times, it will be your responsibility to do the follow-up. This is definitely one of the most important steps to remember regarding what to do after an interview. Sometimes, things get crazy at work or papers and files get misplaced. A gentle follow up after a while isn’t going to hurt anything or make you seem desperate (something many jobseekers have expressed concern with). If you were told there was a specific date to expect a response, wait until that date has passed and then reach out.

Managing what to do after an interview is more about keeping your nerves in check and following up correctly. Don't ever skip these steps because you think the job interview went terribly and there's no point, you really never know. We've heard from so many individuals that thought the interview was awful and still got an offer.

Be optimistic

In case your interview lasts more than 45 minutes or, if there is a detailed discussion about your salary and preferred location of joining. Also in case you are asked to wait or an indication about the second interview has been given, you do stand a chance. All the above signs means that things have gone well and you have left a favorable impression on the interviewers.

Getting feedback

Even if you are not hired, try to get some feedback from the company. This may be of great help for future interviews. Sometimes, not being hired has no reflection on your capabilities so do not lose heart; keep your spirits up because tomorrow is another day.

Analyze the interview

It is one of the most important exercises to do post an interview. Sit down for a few minutes and write down the questions that you were asked during the interview. Furthermore, assess your responses to these questions and make points of things that you forgot to mention or would have said in a better way. This will help you prepare better for future interviews.

Inform your references

Inform your references beforehand that they might receive a call from the company you interviewed for. Speak to them about the position and emphasize on the points that you would like to be recommended for.

Common interview mistakes and ways to avoid them

Remember, for a fresher without experience, the evaluation will mostly rely upon how you conduct yourself through these interviews. Candidates usually make some common mistakes, occasionally knowingly and occasionally unknowingly. Knowingly, because of overconfidence, and unknowingly, because of insufficient preparation. Right here are the most common mistakes candidates make during an interview and let's see how to avoid those mistakes.

Landing late for the interview

The time at which the candidate arrives at the interview location gives an idea about his/her characteristics and punctuality. When you reach the interview location quarter hour before the scheduled time, you will be giving an impression to the panel and the HR that you value the job and you have great time management skills.

Reaching late at the interview location will show you in a poor light and that you have poor time management skills. It could be interpreted as a lack of respect towards the organization and to the interview panel.

The best time to arrive for an interview is 15 minutes prior to the scheduled time slot. Once you reach the interview location, first let the concerned party, that is, the HR department know about your arrival. Make use of the time prior to the interview to remember points you have prepared for the interview and to keep your CV and other documents in order.

The dissimilarity between CV and info provided on the application form

After collecting your CV from online job portals or recruitment agencies, interviewers scrutinize the CV, and if indeed they think it is good, they'll give you a call for the interview. Before attending an interview, an organization asks you to complete a job form which includes all of your past experiences, education details, previous employer history, contact detail, and more.

Make sure the info is accurate in the CV like schooling, education, employment background with the right joining and relieving date, address, phone number, and so on.

Sometimes, it really is difficult to recall everything about past encounters and education. Therefore, the great thing while filling a job application form is to refer to your latest CV, so that you can deliver consistent information to the organization, which assists in creating a faithful image to the panel.

Inappropriate outfit

The candidate will end up being judged first based on the attire. Regardless, for which company the interview is planned, one needs to think and plan what things to wear and make sure that it gives a professional outlook. Based on the organization and job profile, you should decide what things to wear to create the right impression for the job profile.

For example, for a Blue chip company, you can wear business casuals, and for a job interview at a small-established organization, you can go in casual attire.

Body language

The body language throughout the interview discussion could make or break the interview.

The limp handshake/Slouchier, not making a firm handshake and slouching during the interview can make you appear like you aren't assured or lack confidence. Practice shaking hands until you get a firm grip and smile while shaking hands. Sit upright together with your shoulders back through the interview. This enables you to be even more attentive and alert.

Too relaxed, leaning back in the chair in the interview discussions makes the recruiter think that that you may not be interested in the role.

Nervousness, playing with your hair, looking at the hands or around the room will make them think that you aren't confident. At the same time do remember, not to stare at them either, and be sure you look at the interview panel when you respond.

Talkative nature

Many candidates are actually talkative by nature. Those candidates search for someone to talk, they don’t think very much about others, whether others are hearing, or are others getting bored? The talkative applicants do not do it intentionally. Because of their nature, once they start speaking they don’t end. They behave likewise in the interview as well. Once they start, they don't stop and stretch the particular topic longer till the interviewers inform them to stop.

Being talkative is great, but being too talkative may irritate the panel in a way that they feel never to put any more queries to you. Try to control the habits/emotions and promise yourself that I'll limit my talk to only what is required for the occasion.

Paying less attention

Ensure that before arriving at the interview, you have made up your mind and prepared for the interview which means that you'd slept well a day prior to the interview, prepared all questions and answers linked to the job profile, personalized your CV based on the job requirements.

In case you choose an interview timeslot with adequate preparation and less sleep, your mind may not be able to concentrate on the questions asked in the interview, and you'll end up answering wrongly or may answer only a part of the question. This kind of behavior enables you to look bad before the panel and enforce companies to speculate that in case you are uncomfortable in giving the initial interview, you won't be comfortable in attending subsequent interview rounds.

The key is to remain attentive and focused through the interview to ensure that you have a successful discussion with the interview panel.

The HR round does not have the right or wrong response. All these ideas and strategies ensure that you aren't rejected for factors that are in your control. It is necessary to be on your own and present the very best side in this interview.

Conclusion

Do a search on Google before you attend the interview and understand the products and services the company offers. Remember the names of the panelists so that you can address them during the interview discussions.

Even if you have a job, continue with your networking. It is a way to present yourself and also to find something better. Network with stakeholders in your social network, visit professional conferences and fairs. This will increase your chances of job hunting.

After the interview gets over, don’t forget to show your gratitude by stating many thanks with a smile and a good handshake. By thanking the panel, you may also state I am happy to be right here, and I look forward to hear from you once again. Your gratefulness may lead to impressing the panel.

In this chapter, we covered aspects related to the interview preparation process in a step-wise manner, that is, the guidelines to follow before, during, and after an interview. In the next chapter, we will learn about data structures and algorithms and we will cover aspects, including core concepts, collections, algorithms, and miscellaneous topics.

CHAPTER 4

Data Structures and Algorithms

This chapter presents an exhaustive question bank on data structures with special emphasis on practical scenarios and business cases for key topics. This chapter and the next chapter covers all key domains, including data structures, OOPs, DBMS, OS, and methodologies and processes, and trending programming languages.

This chapter covers the topic of data structures in the programming languages, which essentially provides a number of data types and methods that can be performed on these data types. This chapter covers the core concepts, principles, collections, algorithms, programming standards, and other key topics. A data structure is a concrete implementation of the common data types, that is, linked lists, stacks, queues, maps, and many more, which are covered in this chapter. Different data structures are suited for different application requirements, and a select few are specialized for specific tasks. For example, compilers leverage hash tables to research identifiers where-as RDBMS typically leverage B-tree indexes for data retrieval.

Concepts

What is a data structure?

Data structures provide functions used to manipulate large size data efficiently; for example, internet indexing solutions or databases. Effective data structures are fundamental to architecting optimal applications. Design strategies and programming languages emphasize on these data structures instead of algorithms, as they are critical factors in software applications. Data structures are leveraged to structure the storage and retrieval of information stored in both primary and secondary memory.

As mentioned earlier, anything that stores data can be called as a data structure ; hence, an integer, Boolean, char, float, and more. Each of them is a data structure and so are referred to as Primitive data structures . Additionally, there are some complex data structures, which are accustomed to storing large and linked data. Some examples of Abstract data structure are as follows:

Linked list

Graph

Tree

Queue

Stack

These data structures allow you to perform numerous operations about data. A data structure is selected based on the type of procedure is needed for the application. The following diagram depicts all the primitive and non-primitive data structures supported by all the key programming languages. In the diagram, the last level data structures are the specialization of top-level structures; for example, stacks and queues are linear lists whereas graphs and trees are non-linear lists:

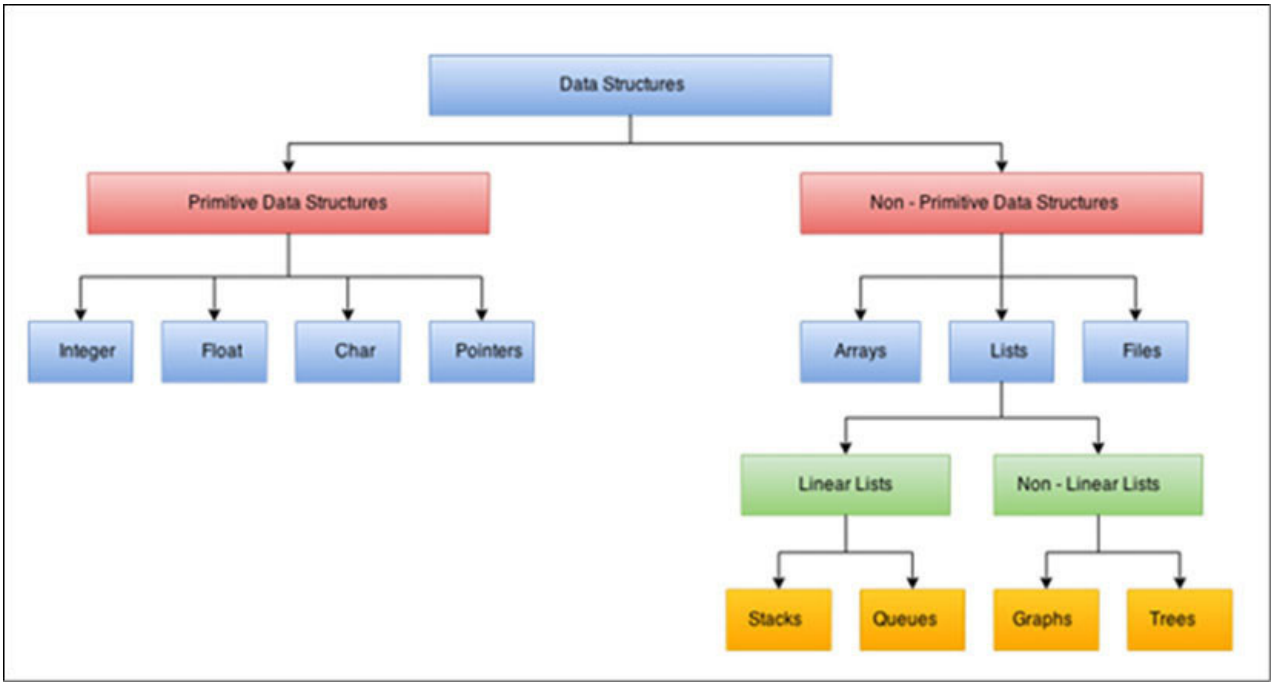


Figure 4.1: Family of data structures

A data structure is a way to structure data that considers the data storage and their relationship. Advance abilities about the relationship between data items allow you to design effective algorithms for manipulating data.

What are the various operations that can be performed on different data structures?

Various operations can be performed on different data structures, which are as follows:

Insertion: Add a new data item in the given collection of data items.

Deletion: Delete an existing data item from the given collection of data items.

Traversal: Access each data item exactly once so that it can be processed.

Searching: Find out the location of the data item if it exists in the given collection of data items.

Sorting: Arranging the data items in some order, that is, in ascending or descending order in case of numerical data and in dictionary order, in case of alphanumeric data.

Where are data structure leveraged?

Data structures can be leveraged in the following areas:

Operating system (OS)

Compiler design

DBMS/RDBMS

Numerical analysis

Statistical analysis package

Graphics

Simulation

Chat-bots, Machine Learning, Artificial Intelligence

RDBMS such as arrays

Network data model such as graphs

Hierarchical data model such as trees

What are generic collections?

Generics contains interfaces and classes that define generic collections, which allow users to create strongly typed collections that provide better safety and performance than non-generic strongly typed collections. System Collections namespace provides numerous classes and interfaces. For example, List, ArrayList, ICollection, Stack, IDictionary, and IEnumerable . Generic offers type-safety to class(es) at compile time. While creating a data structure leveraging generics, one doesn’t need to specify the data type at declaration time. System.Collections.Generic namespace contains all the generic collections. Code that uses generics has many benefits over non-generic code: Stronger type checks at compile time. A Java compiler applies strong type checks to generic code and issues errors if the code violates type safety. Fixing compile-time errors is easier than fixing runtime errors, which can be difficult to find.

What is the difference between collections and arrays?

Collections: The collection can store and manipulate data items of different types. Collections don't have a fixed or predefined size and can be adjusted at run time, as per the application requirements.

Arrays: One needs to declare the size of the array at declaration time and it cannot be adjusted at runtime. Elements of an array consist of the same data types.

What is the set interface?

The Set interface is leveraged to represent a collection, which will not contain duplicate elements. The Set interface extends the Collection interface. It specifies that an instance of the set will not contain duplicate elements.

Classes that implement the set interface must ensure that no duplicate data items are added to set collections. This means that the two elements, ex1 and ex2, can't be in a set if ex1.equals(ex2) is true.

When to use the comparator and comparable in code?

The Comparable interface is simpler and requires less effort. The following are the key characteristics:

Class implements the Comparable classes.

Class defines the public method compareTo(Object o) .

No arguments are passed to the TreeMap or TreeSet constructor.

There is only one comparison method leveraged each time.

The Comparator interface is flexible but requires more effort. The following are the key characteristics:

One may create classes implementing the comparator as per the application requirements.

One can sort TreeMap or TreeSet differently for each Comparator .

TreeSet or TreeMap leverages the comparator in the constructor as per application requirements.

What is a linked list and types?

A linked list is a linear data structure (like arrays) where each element is a separate object. Each element (that is a node) of a list comprises two items: the data and a reference to the next node. There are three types of lists, which include the following:

Singly linked list: In this type of linked list, every node stores an address or reference of the next node in the list and the last node has the next address or reference as NULL . For example, 20->30->40->NULL .

Doubly linked list: Here, there are two references associated with each node. One of the reference points to the next node and one to the previous node. For example, NULL<-20<->30<->40->NULL .

Circular linked list: This list is a linked list where all nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be a singly circular linked list or doubly circular linked list. For example, 20->30->40->20. [The next pointer of the last node is pointing to the first.]

The following diagram depicts the three different types of data structures and how one can transverse from one element to another:

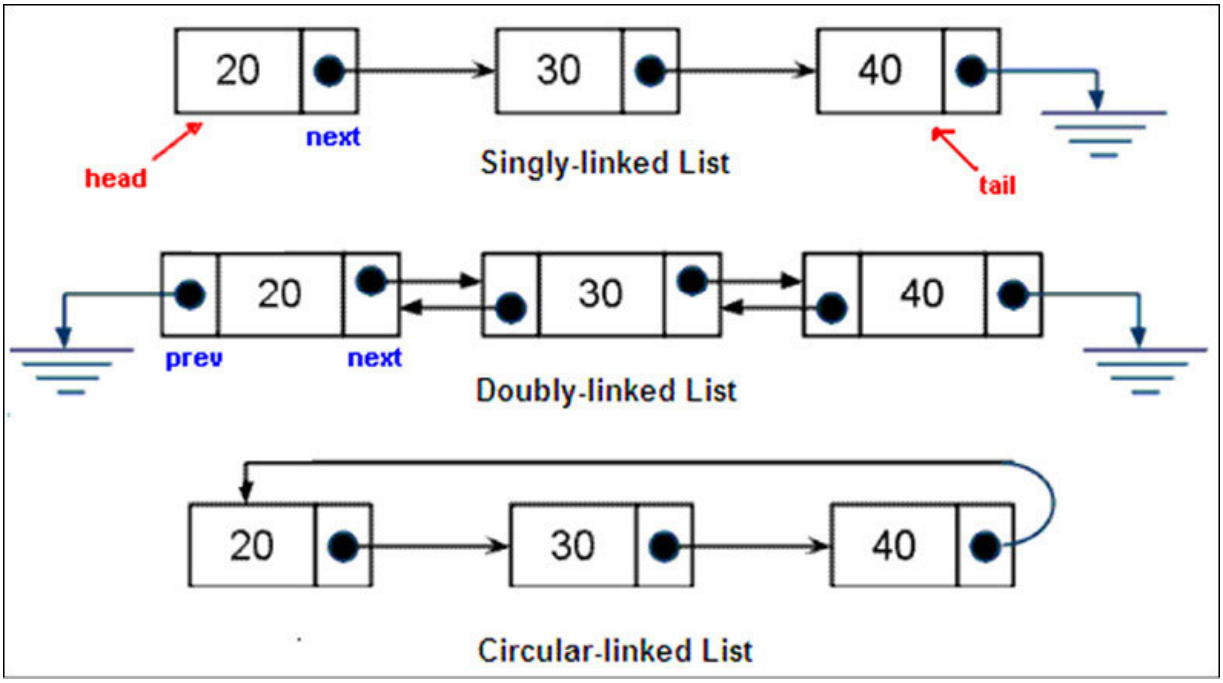


Figure 4.2: Singly vs doubly vs circular linked list

Which pointer to select if one is leveraging the C language to implement a heterogeneous linked list?

The heterogeneous linked list stores various data types, and a pointer needed for connecting them cannot be ordinary pointers and hence, one has to select void pointers. Void pointers have the ability to store pointers to any data types since it is a generic pointer type.

Differentiate a file structure from the storage structure.

Actually, the key difference is the memory area that is being accessed. When dealing with the structure that resides in the main memory of the computer system, this is referred to as a storage structure. When dealing with an auxiliary structure, we refer to it as file structures.

What is a Queue, how it is different from stack and how is it implemented?

A Queue is a linear structure that follows the order first in first out (FIFO) to access elements. Mainly, the basic operations on queue are enqueue, dequeue, front, andrear. The difference between stacks and queues is in removing the element. In a stack, we remove the most recently added item; in a queue, we remove the firstly added item. Both Queue and Stack can be implemented using arrays and linked lists. The following diagram depicts the stack and queue operations for adding and removing elements:

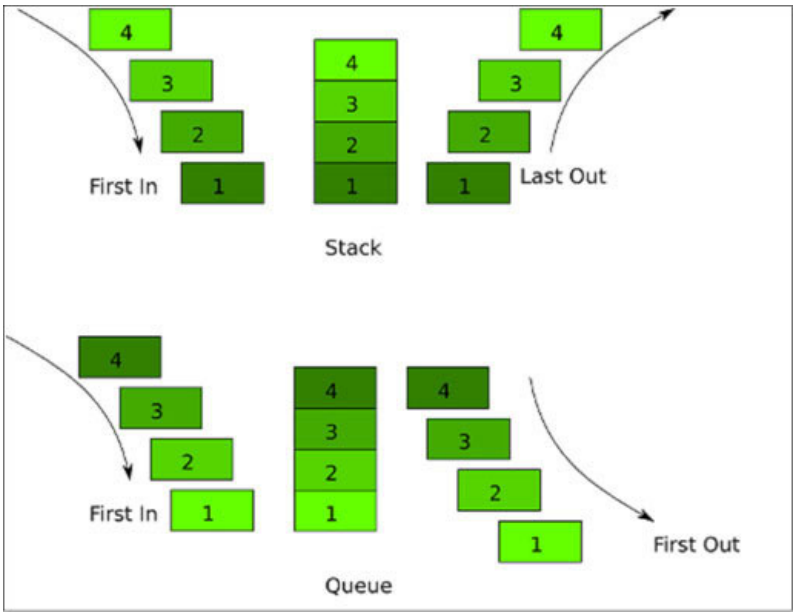


Figure 4.3: Stack vs Queue data structure operations

How much minimum number of queues is needed to implement a priority queue?

To implement a priority queue, two queues are required in the application. One queue is leveraged for storing of data and the other queue is required for storing.

Collections

What is the objective for collection classes?

Collection classes provide different functions to operate on collections, maps, and datastructures, enabling synchronization and making the classes read-only.

Compare a LinkedList and an ArrayList?

An ArrayList leverages a dynamic array, whereas LinkedList leverages a doubly linked list.

Data manipulation functional is faster with a LinkedList compared to an ArrayList .

An ArrayList can only be leveraged as a list, but a LinkedList can be leveraged as a list and a queue.

The following diagram depicts how data elements are transversed and stored in the linked list and array list:

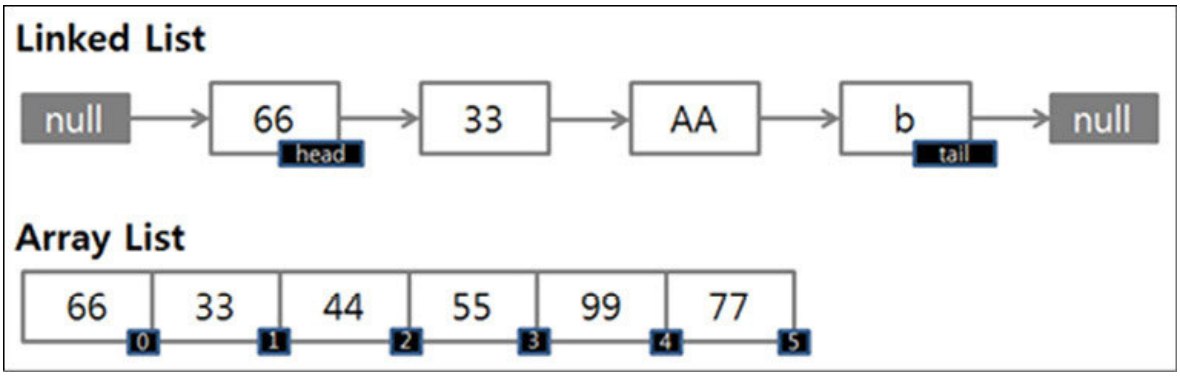


Figure 4.4: Linked list vs array list

What is a Vector class?

The Vector class is comparable to ArrayList with regards to accessing the items in a collection. Once an object of the vector is created, its size can increase or decrease whenever we add or remove items in a Vector . It is synchronized by default.

What is a Stack class?

A Stack class represents Last In First Out (LIFO) queue of objects. The data items in the stack can only be accessed from the top of the stack. One can insert, remove or retrieve a data item from only the top of the stack. The following diagram depicts the push and pop operations on the Stack data structure. The following diagram depicts the state of the stack class after every operation on the right-hand side of it:

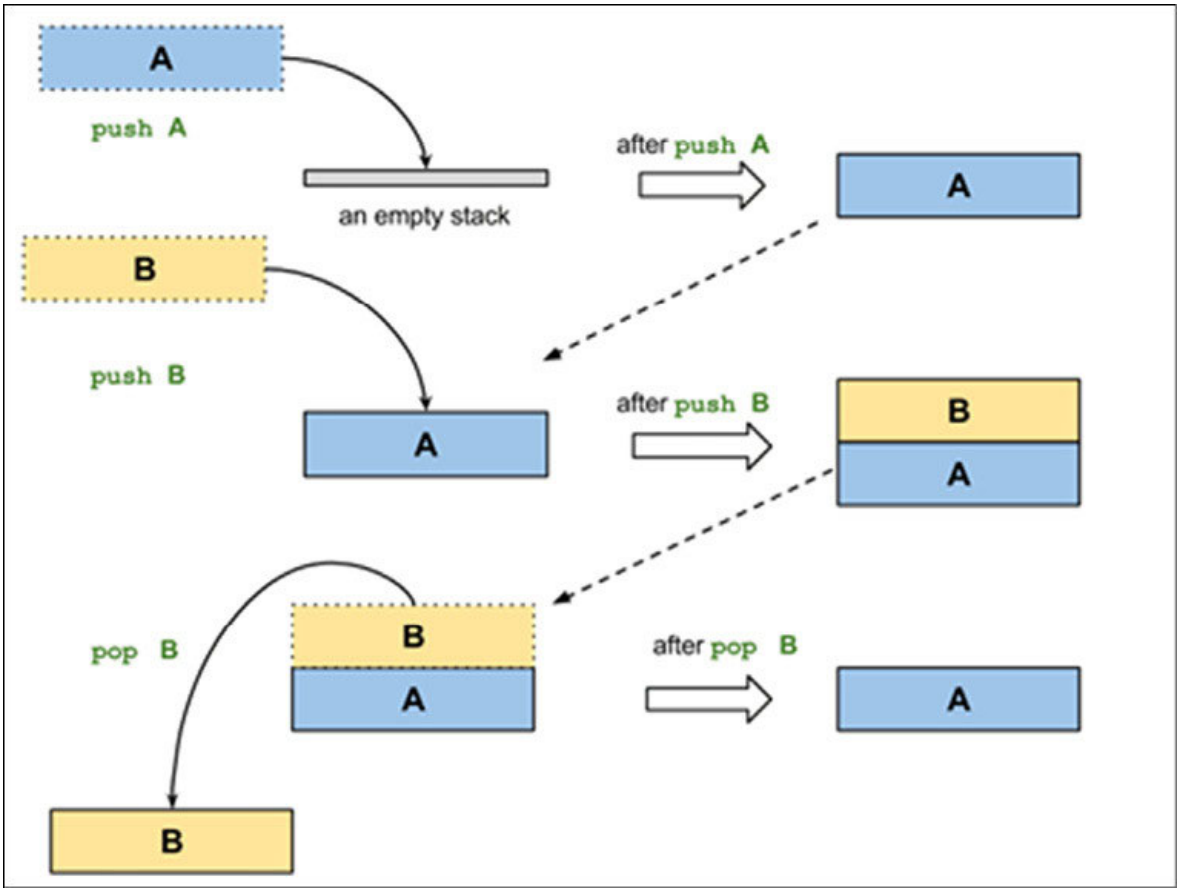


Figure 4.5: Stack operations

What is a Map interface?

The Map interface is a data structure that maps keys and values. The map will not support duplicate keys and each key maps only to one value. This Map interface is not an extension of the Collection interface. Instead, the interface begins its interface hierarchy via the Map . The interface facilitates mapping between the key and value pairs, without duplicate keys. In the map, the keys are like indexes compared to a list, where indexes are integers. The following diagram depicts the different specializations of the Map class, which are based on predominately-different algorithms

that are leveraged for map hashing, that is, storing and retrieving data elements.

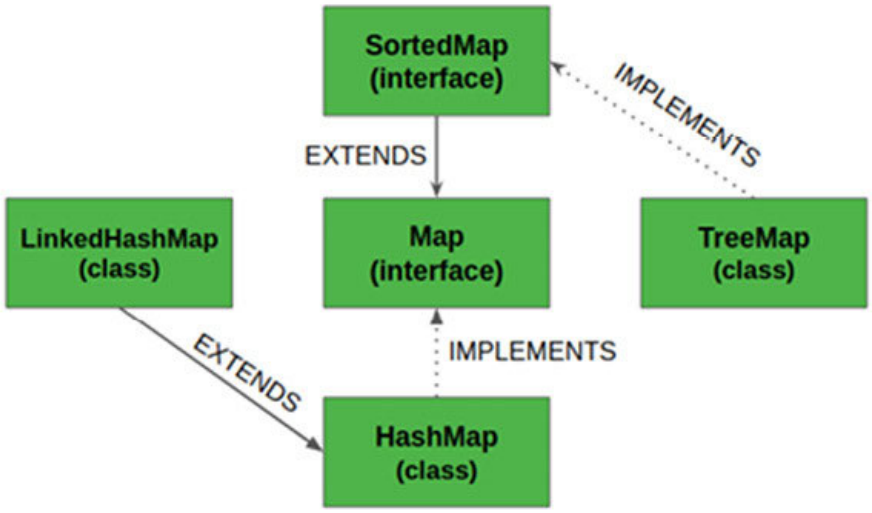


Figure 4.6: Map class hierarchy

Describe HashMap and TreeMap.

HashMap and TreeMap are concrete implementations of the Map interface. Both need key-value pairs and the keys are unique. HashMap is normally effective for locating a value, inserting a mapping, and deleting a mapping. TreeMap implements SortedMap and is usually very effective at traversing the keys in a sorted order. HashMap permits storing null as both keys and values. In terms of performance, TreeMap is usually slower than HashMap . In TreeMap , a Comparator object is utilized to choose the order in which the keys are sorted.

What are limitations of an array?

Arrays do not grow as applications demand. An array does not provide searching, sorting, inserting and deleting operations.

What is a collection framework?

A collection framework offers a generic group of methods and interfaces to take care of collections. It may be summarized as follows:

Interfaces that characterize common collection types.

Classes providing implementations of the interfaces.

Abstract classes which are leveraged as a starting place for custom collections and which are expanded by the JDK implementation classes.

Algorithms providing behaviors common to collections, that is, search, sort, iterate, and many more.

To gain access to every item of a collection, the framework provides the Iterator interface. The advantage of the iterator interface is that it provides a clear standard methodology to gain access to the data elements in a collection, one at a time. The following diagram depicts the hierarchy of the collection classes that one can leverage in programming constructs. The leaf classes are specializations of the parent classes; for example, the List class has specialization in Arrarylist , Vector , and LinkedList . Similarly, Queue has specialization in Dequeue and the priority queue:

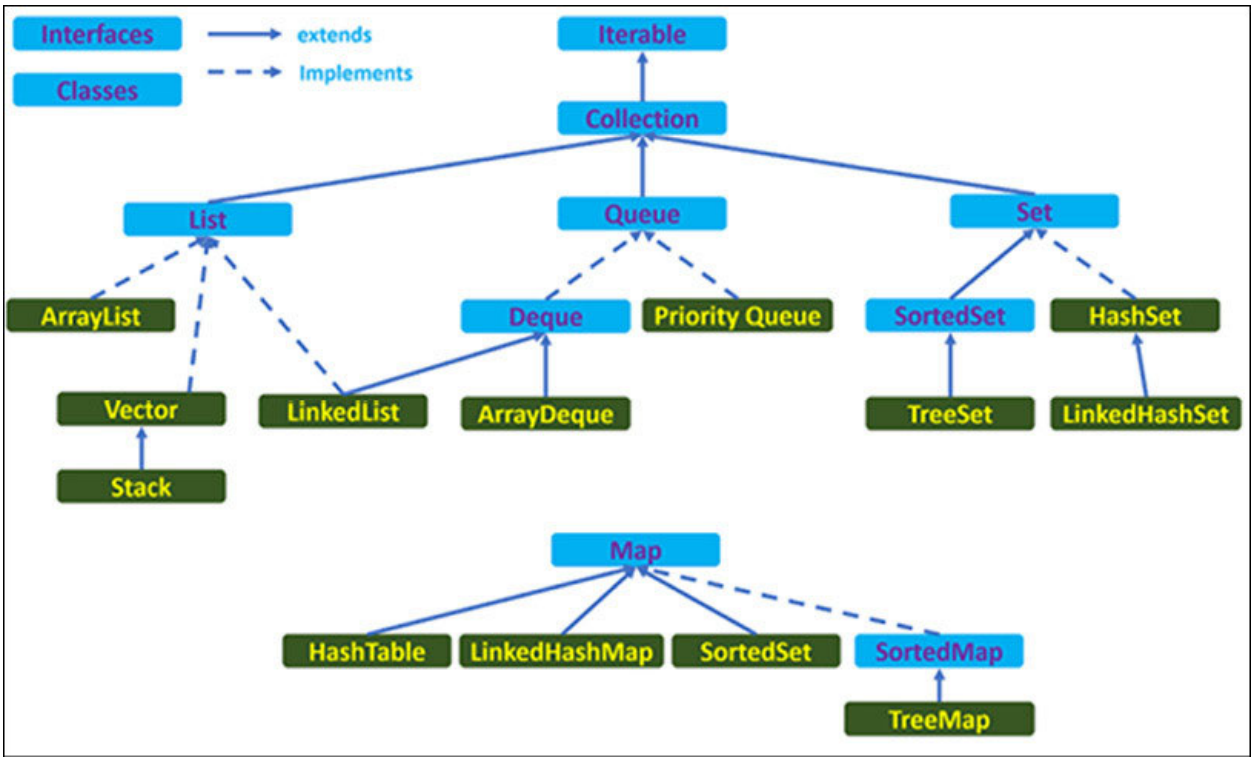


Figure 4.7: Collection class hierarchy

Explain HashSet, LinkedHashSet, and TreeSet.

HashSet, TreeSet, and LinkedHashSet are the three general-purpose imple-mentations of the set interface supplied by the Java framework.

HashSet : This uses hashmap to store elements. This is an unsorted collection, unordered Set . This can be selected when the order of the data elements are not important. This gives better performance than LinkedHashSet and TreeSet .

LinkedHashSet : This uses linkedhashmap internally to store elements. This is an ordered collection. The data elements are linked to each other by leveraging the doubly linked list. This will maintain the list in the order in which they were inserted. The performance is moderately good and almost similar to HashSet .

TreeSet : This is a sorted set collection. The data elements in this set will be in ascending order by default. You can also define a custom made order utilizing a comparator shared as a parameter to the constructor. The performance is not comparable to HashSet and LinkedHashSet as it has to sort the entire collection during insertion and removal.

The following table compares TreeSet , HashSet , and LinkedHashSet structures on various parameters.

	TreeSet	HashSet	LinkedHashSet
Duplicates	No	No	No
Thread safety	No	No	No
Speed	Slow	Faster	Medium
Order	Sorted order	No order	Insertion order
Null values	No	Allow	Allow
Implementation	By NavigableMap	HashMap	LinkedList & HashSet

Figure 4.8: Comparing HashSet, LinkedHashSet and TreeSet

Which data structure to leverage among HashSet, LinkedHashSet, and TreeSet?

HashSet is an excellent selection for data sets if the order of data items in a set is not important. In case ordering is critical, then LinkedHashSet or TreeSet are the excellent choices. Nevertheless, LinkedHashSet or TreeSet includes an additional swiftness and space overhead.

Iteration over a LinkedHashSet is normally quicker than iteration over a HashSet .

Tree-based data structures become slower as the number of data elements increase.

LinkedHashSet and HashSet usually do not signify their data elements in a sorted order.

TreeSet provides additional features such as finding the initial and last data elements in a set.

What is the Enumeration interface?

The Enumeration interface helps us to loop through the objects in a collection. To check whether an element is present in a collection, the interface provides the MoreElements() method. It'll return a Boolean true if a data element exists.

If hasMoreElements() returns true and we want to fetch the next object in the collection, we leverage the nextElement() method. If there are no more objects that exist in the collection and if we invoke the nextElement() method, the runtime will throw the NoSuchElementException exception.

Explain the Iterator interface.

An iterator is a special entity that delivers a mechanism to gain access to data-elements in the collection sequentially. The Iterator class implements one of the two interfaces: Iterator or ListIterator . An Iterator is comparable to the Enumeration interface; iterators are different from enumerations in the following ways:

Iterators permit the code to remove data elements of a collection during the iteration. Method names utilized by the iterator are shorter in comparison to the Enumeration interface.

The hasNext() method of the Iterator interface returns true, if the iteration has more elements. To obtain the next object in the iteration, call next. Call the remove() method (optional) to remove the last element returned by the iterator.

The following diagram depicts the context in which the iterator interface is leveraged in conjunction with a collect class:

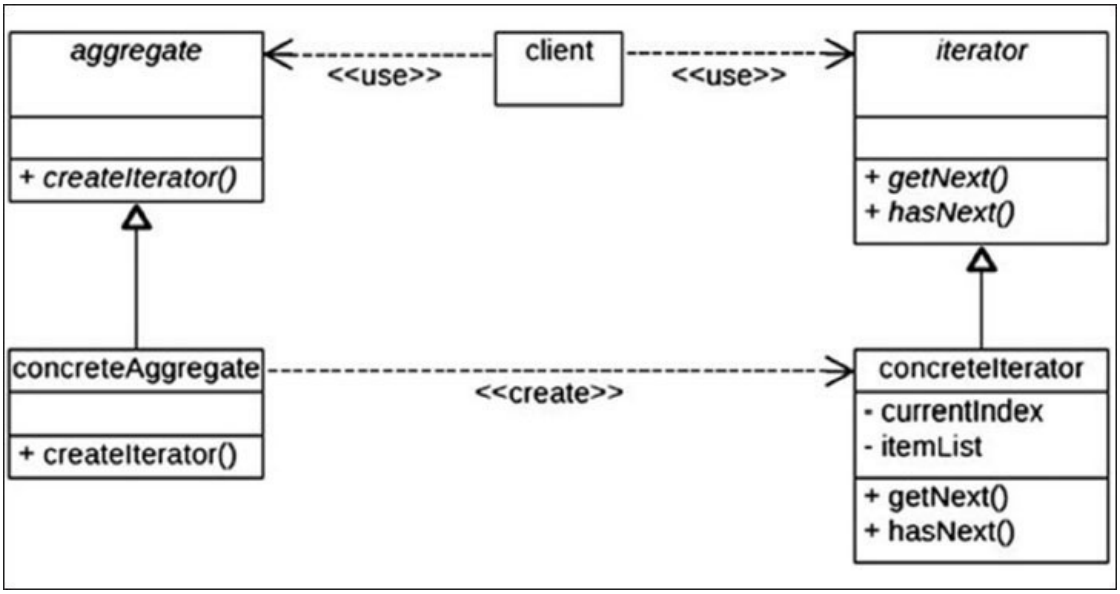


Figure 4.9: Iterator interface context

What is the SortedSet interface?

The SortedSet extends the Set interface, which ensures that the elements of a set are sorted. An example of a SortedSet interface is TreeSet . One can utilize an iterator to fetch the sorted elements in the SortedSet .

What is a list interface?

The List interface facilitates creating a list of data items. The List interface extends the Collection interface.

A list enables storing duplicate data items in a collection. The list allows you to specify where the data item can be stored. The data items can be accessed by an index.

Define the circular list.

In the linear list, the next field of the last node contains a null pointer. When the next field in the last node contains a pointer back to the first node, it is called a circular list. There are two variations of a circular linked list: one variation that is the circular single linked list. Each node stores two elements: one element is the data and the second element is a pointer to the next forward element. In the doubly circular linked list, each node stores three elements: one is the data and the other two are the pointers to the previous and next node of the list. The following list describes the features of the different circular lists:

Singly circular list:

Traversing in one direction

Program will be lengthy and needs more time to design

While deleting the node, its pre-processor is required that can be found by traversing from the beginning of the list

Less space per node is needed

Elementary operations are less expensive

Bit difficult to manipulate

Doubly circular list:

Traversing can be done in both directions

Efficient and lean program can be written and hence the design is easier

While deleting the node, its pre-processor is part of the existing node and hence, no need to traverse the entire list

More space per node is needed

Elementary operations are more expensive

Easier to manipulate

Please refer to the following diagram for the depiction of these two variations of the list:

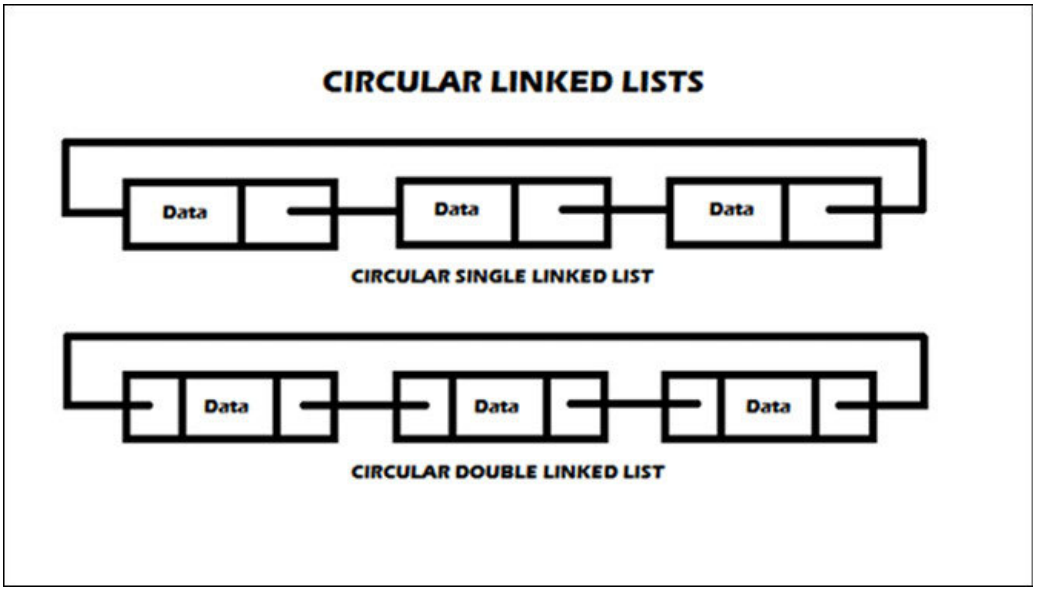


Figure 4.10: Circular singly vs doubly LinkedList

Define the double linked list.

The double linked list is a collection of data elements called nodes, where each node is divided into three parts: an info field that contains the information stored in the node, a left field that contains a pointer to the node on left-hand side and the right field that contains a pointer to the node on right-hand side. The following diagram depicts the doubly linked list date structure operations:

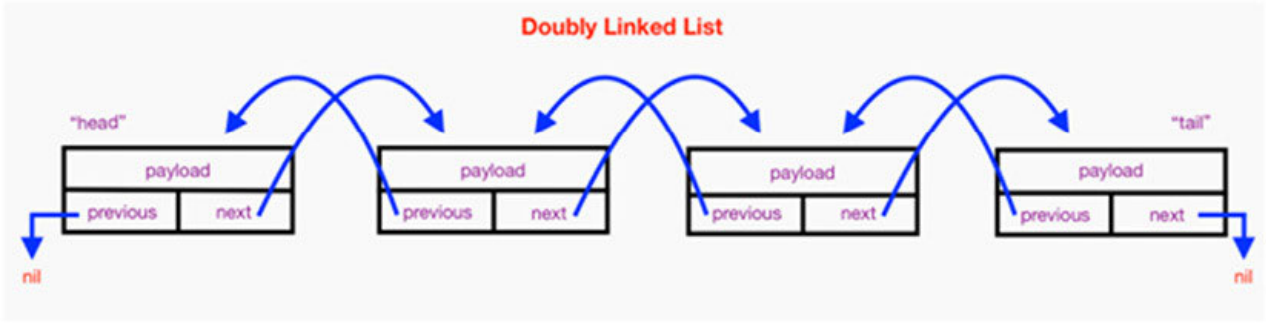


Figure 4.11: Doubly linked list structure

Explain the terminology LIFO and FIFO.

LIFO stands for last in first out. This process describes how the data is accessed, stored and then retrieved. So the latest data that is stored in the database can be extracted first.

FIFO in data terminology stands for first in first out. This process defines or depicts how the data is stored, inserted and accessed in a queue. Within this process, the data that is inserted at the beginning of the queue will only be extracted or accessed first. The following diagram depicts the difference between the LIFO and FIFO operations:

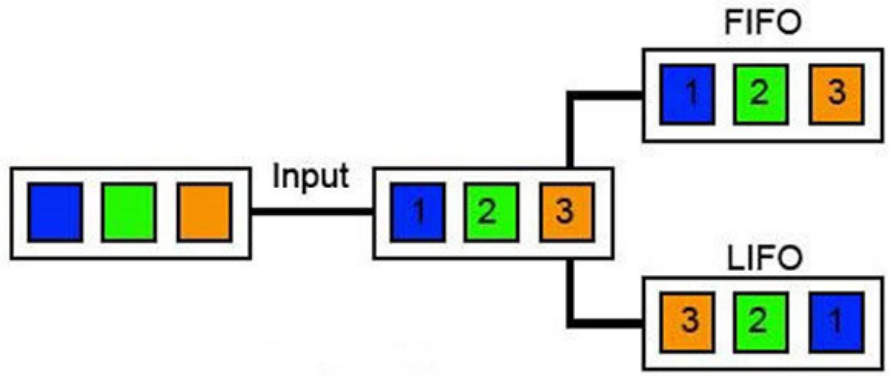


Figure 4.12: LIFO vs FIFO operations

List all the advantages of a linked list.

The important aspect or advantage of a linked list is that it is the perfect data structure where the data can be modified very easily. Also, it doesn't matter how many elements are available on the linked list.

Explain the main difference between a PUSH and a POP.

The two main activities, that is, pushing and popping applies to the way data is stored and retrieved in an entity. So if you check in detail, a push is nothing but a process where data is added to the stack. On the contrary, a pop is an activity where data is retrieved from the stack. When we discuss the data retrieval, it only considers the topmost available data.

Define the advantages and disadvantages of the heap compared to a stack.

The advantages of a heap compared to a stack are listed as follows:

A heap is more flexible when compared to a stack.

The memory space of the heap can actually be allocated and de-allocated as per the need.

On the contrary, disadvantages of a heap compared to a stack are listed below:

The memory of the heap is slower when compared to the memory of the stack.

Explain how the new data can be inserted in the tree.

The following are the steps that you need to follow to insert the data into the tree:

First of all check whether the data is unique or not (that is, check whether the data that you are going to insert doesn't already exist in the tree).

Then, check whether the tree is empty. If the tree is empty, then all you need to do is just insert a new item into the root. If the key is smaller to that of a root's key, then insert that data into the root's left subtree or otherwise, insert the data into the right-hand side of the root's subtree.

Can you tell me the minimum number of nodes that a binary tree can have?

A binary tree is allowed or can have a minimum zero nodes. Further, a binary tree can also have one or two nodes.

Explain a little bit about the dynamic data structure.

The nature of the dynamic data structure is different to the standard data structure. The word dynamic data structure means that the data structure is flexible in nature. As per the need, the data structure can be expanded and contracted. Thus, it helps the users to manipulate the data without worrying too much about the data structure flexibility.

Can you tell me the minimum number of queues that are needed to implement a priority queue?

The minimum number of queues that is needed is two. Out of which, one queue is intended for sorting priorities and the other queue is meant for the actual storage of data.

Explain a Dequeue.

A Dequeue is nothing but a double-ended queue. Within this structure, the elements can be inserted or deleted from both the sides.

Explain a graph.

A graph is nothing but a type of a data structure, which has a set of ordered pairs. In turn, these pairs are again acknowledged as edges or arcs. These are used to connect different nodes where the data can be accessed and stored based on the needs.

Which data structure should be used for implementation of the LRU cache?

A Queue that is implemented using a doubly linked list. The maximum size of the queue will be equal to the total number of frames available (cache size). The most recently used pages will be near the front end and least recently pages will be near the rear end.

A Hash with a page number as the key and address of the corresponding queue node as a value.

How to implement a stack using a queue?

A stack can be implemented using two queues. Let the stack to be implemented be s and queues used to implement be q1 and q2 . Stack s can be implemented in the following two ways:

Method 1 (By making the push operation costly)

Method 2 (By making the pop operation costly)

How is an array different from a linked list?

The size of the arrays is fixed; linked lists are dynamic in size. Inserting and deleting a new element in an array of elements is expensive, whereas both insertion and deletion can easily be done in linked lists. Random access is not allowed in linked lists. Extra memory space for a pointer is required for each element of the linked list. Arrays have better cache locality that can make a pretty big difference in the performance.

Given a choice between ArrayList and LinkedList, which one would you use and why?

ArrayList are implemented in the memory as arrays and hence it allows fast retrieval through indices but they are costly if new elements are to be inserted in between other elements. LinkedList allows constant-time insertions or removals using iterators, but only sequential access of elements. The following lists the various operations that can be performed on the ArrayList data structure:

Retrieval: If elements are to be retrieved sequentially only, a linked list is preferred.

Insertion: If new elements are to be inserted in between other elements, a linked list is preferred.

Search: Binary search and other optimized way of searching are not possible on a linked list.

Sorting: Initial sorting could be a pain but lateral addition of elements in a sorted list is good with a linked list.

Adding elements: If sufficiently large elements need to be added very frequently, a linked list is preferable as elements don't need consecutive

memory location.

Algorithms – Sorting and Searching

What options are available for storing sequential files?

Straight merging

Natural merging

Polyphase sort

Distribution of initial runs

How can we order/sort objects in collections?

To provide ordering/sorting of the objects, Java API provides two interfaces Comparable and Comparator :

The Comparable interface is in java.lang .

The Comparator interface is in java.util .

How are elements sorted in Java collections?

Elements can be sorted in two ways. One way is usually to leverage the comparable interface. Objects can be compared using the compareTo() method. This process is actually a natural ordering.

The next alternative is to specify a comparator for the data items in the set. In the event if the class does not implement the Comparable interface, or you don't want to leverage the compareTo() function that implements the Comparable interface, then the approach is known as order by the comparator.

In which of the methods sorting is not feasible ? (Insertion, Selection, Exchange, Deletion)

Sorting isn't feasible in the deletion operation. Leveraging insertion one can perform the insertion sort; leveraging selection one can perform the selection sort; leveraging exchange one can perform the bubble sort and leveraging other identical sorting methods. The sorting function cannot be carried out by leveraging deletion.

What is a recursive function?

A recursive function is one which calls itself, directly or calls a function that in turn calls it. Every recursive function follows the recursive properties: base criteria where functions stop calling themselves and the progressive approach where the functions try to meet the base criteria in each iteration. An important application of recursion in computer science is in defining dynamic data structures such as lists and trees.

When is a binary search best applied?

A binary search is an algorithm that is best applied to search a list when the elements are already in an order or sorted. The list is searched starting in the middle such that if the middle value is not the target search key, it will check to see if it will continue the search on the lower half of the list or the upper half. The split and search will then continue in the same manner. The following diagram depicts the binary search algorithm and explains it in a step-wise manner as to how the algorithm works:

Figure 4.13: Binary search algorithm

What is an ordered list?

An ordered list is a list in which each node's position in the list is determined by the value of its key component, so that the key values form an increasing sequence as the list is traversed.

What is merge sort?

Merge sort is a divide-and-conquer approach for sorting the data. In a sequence of data, adjacent ones are merged and sorted to create bigger sorted lists. These sorted lists are then merged again to form an even bigger sorted list, which continues until you have one single sorted list.

Differentiate between NULL and VOID.

Null is a value, whereas Void is a data type identifier. A variable that is given a Null value indicates an empty value. The void is used to identify pointers having no initial size.

Which sorting algorithm is considered the fastest?

There are many types of sorting algorithms: quick sort, bubble sort, balloon sort, radix sort, merge sort, and many more. Not one can be considered the fastest because each algorithm is designed for a particular data structure and data set. It would depend on the data set that you would want to sort.

What is a bubble sort and how do you perform it?

A bubble sort is a sorting technique that can be applied to data structures such as arrays. It works by comparing adjacent elements and exchanges their values if they are out of order. This method allows the smaller values bubble to the top of the list, while the larger values sink to the bottom.

In step 1, the algorithm will swap elements 7,4 and 7,5 and 7,2

In step 2, it will swap element 5,2

In step 3, it will swap 4,2

These process steps will ensure that the smaller elements are on the left-hand side of the collection and are sorted through the three iterations of the algorithm. The following diagram depicts the bubble sort process in a steps wise manner:

Figure 4.14: Bubble sort algorithm

How does the selection sort work?

The selection sort works by picking the smallest number from the list and placing it in the front. This process is repeated for the second position towards the end of the list. It is the simplest sort algorithm. The following steps are performed:

Number 1, which is the lowest number is selected and placed in the front of the list, which is the first number in the list.

The next number 2 is the next lowest number selected and placed in front of the earlier smaller number, which is 1.

The next number 3 is the next lowest number selected and placed in front of the earlier smaller number, which is 2.

The next number 4 is the next lowest number selected and placed in front of the earlier smaller number, which is 3.

The following diagram depicts the preceding steps:

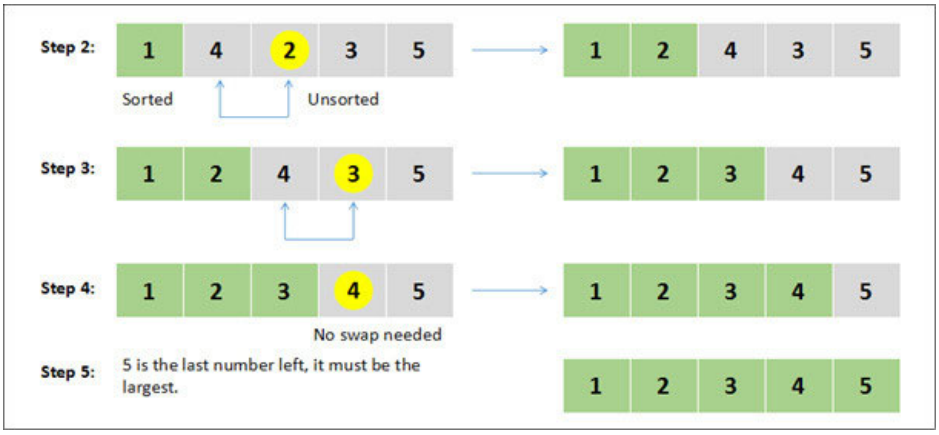


Figure 4.15: Selection sort algorithm

How do you search for a target key in a linked list?

To find the target key in a linked list, you need to apply a sequential search. Each node is traversed and compared with the target key, and if it is different, then it follows the link to the next node. This traversal continues until either the target key is found or if the last node is reached.

List all the different sorting algorithms that are available and state which sorting algorithm is considered as the fastest.

The list of all the sorting algorithms isas follows:

Quick sort

Bubble sort

Balloon sort

Radix sort

Merge sort

Out of the preceding sorting options, none of the sorting algorithms can be tagged as the fastest algorithm because each of these sorting algorithms is defined for a specific purpose. Based on the data structure and data sets available the sorting algorithms are used.

Miscellaneous

What data structures are leveraged to perform recursion?

The stack is leveraged for recursion due to its LIFO property. A stack remembers its caller and therefore knows who to return when the function execution ends. Recursion utilizes the system stack for storing the return addresses of the function calls.

Each recursive function has its equivalent iterative (non-recursive) function. Even when such comparative iterative procedures are created, the explicit stack is leveraged.

List application of tree data-structure

Manipulation of arithmetic expressions

Syntax analysis

Symbol table construction

Which is an efficient data structure in tree construction? (Array, Linked list, Stack, Queue)

A linked list is the efficient data structure in tree construction.

Which data structure is utilized in RDBMS for the internal storage?

B+ tree data structure is leveraged for RDBMS internal storage. As all the data is kept in leaf nodes in B+ tree this makes searching easier. This corresponds to the records that will be stored in leaf nodes.

Which is the simplest file structure among Indexed, Sequential, and Random?

Sequential is the easiest file structure.

What are linear and non-linear data structures? Is a linked list a linear or non-linear data structure?

Linear: A data structure is said to be linear if its elements form a sequence or a linear list. Examples: arrays, linked lists, stacks, and queues.

Non-linear: A data structure is said to be non-linear if the traversal of nodes is non linear in nature. Example: graphs and trees.

It depends on where you intend to apply linked lists. If you based it on storage, a linked list is considered non-linear. On the other hand, if you based it on access strategies, then a linked list is considered linear.

A linked list is a linear according to access strategies. A linked list is a non-linear according to storage.

What is linear searching?

A linear search tries to find an item in a sequentially arranged data type. These sequentially arranged data items known as array or lists are accessible in the incrementing memory location. A linear search compares the expected data item with each of the data items in a list or array. The average case time complexity of the linear search is O(n) and the worst case complexity is O(n2) . The data in target arrays/lists need not be sorted.

Which data structures are applied when dealing with a recursive function?

Recursion is a function that calls itself based on a terminating condition and makes use of the stack. Using LIFO, a call to a recursive function saves the return address so that it knows how to return to the calling function after the call terminates.

Explain a binary search tree.

A binary search tree stores data in such a way that it can be retrieved very efficiently. The left subtree contains nodes whose keys are less than the node's key-value, while the right subtree contains nodes whose keys are greater than or equal to the node's key-value. Moreover, both the subtrees are also binary search trees.

Differentiate between a file and structure storage structure

The key difference between both the data structures is the memory area that is being accessed. When dealing with the structure that resides in the main memory of the computer system, this is referred to as a storage structure. When dealing with an auxiliary structure, we refer to it as a file structure.

What is the advantage of the heap over a stack?

The heap is more flexible than the stack. That's because the memory space for the heap can be dynamically allocated and de-allocated as needed. However, the memory of the heap can at times be slower when compared to that of the stack. The following diagram depicts the memory layout of a program:

Figure 4.16: EA value quadrant

What is a postfix expression?

A postfix expression is an expression in which each operator follows its operands. The advantage of this form is that there is no need to group sub-expressions in parentheses or to consider operator precedence.

How do signed and unsigned numbers affect memory?

In the case of signed numbers, the first bit is used to indicate that it is positive or negative, which leaves you with one bit short. With unsigned numbers, you have all bits available for that number. The effect is best seen in the number range (an unsigned 8-bit number has a range 0-255, while the 8-bit signed number has a range -128 to +127.

In what data structures are pointers applied?

Pointers that are used in a linked list have various applications in the data structure. Data structures that make use of this concept include the stack, queue, linked list, and binary tree.

Do all declaration statements result in a fixed reservation in memory?

Most declarations do with the exemption of pointers. Pointer declaration does not allocate memory for data, but for the address of the pointer variable. The actual memory allocation for the data comes during runtime.

What is Huffman's algorithm?

Huffman's algorithm is used for creating extended binary trees that have minimum weighted path lengths from the given weights. It makes use of a table that contains the frequency of occurrence for each data element.

What is Fibonacci search?

Fibonacci search is a search algorithm that applies to a sorted array. It makes use of a divide-and-conquer approach that can significantly reduce the time needed in order to reach the target element.

Can you explain with an example how can a variable declaration activity consume or affect the memory allocation?

The amount of space or memory occupied or allocated dependson the data type of the variables that are declared. So let's explain the same by considering an example. Let's say the variable is declared as an integer type, then 32 bits of memory storage is allocated for that particular variable. So based on the data type of the variable, the memory space will be allocated.

Conclusion

Data structures are the logical or mathematical arrangement of data in memory. They consider not only the physical layout of the data items in the memory but also the relationships between these data items and the operations that can be performed on these items. The data structure is nothing but an entity where the data is perfectly aligned and it can be manipulated as per the requirement. When we deal with data structure, it is not just about one table of data but it is about different data sets and how well they are aligned with each other and help organize the data.

This chapter covered the question bank for data structure core concepts, collections, algorithms, and critical topics. The next chapter will cover the question bank for the operating system fundamentals, concepts, tools, and processes.

CHAPTER 5

Operating System

An operating system is a program on which application programs are executed and it acts as a communication bridge (interface) between the user and the computer hardware. The main task an operating system carries out is the allocation of resources and services such as allocation of memory, devices, processors and information. The operating system also includes programs to manage these resources such as a traffic controller, a scheduler, memory management module, I/O programs, and a file system.

This chapter covers the question bank for the operating system domain. This includes core concepts, process management, memory and file management, multi-threading and synchronization.

Concepts

What is an operating system?

An Operating System (OS) is normally a layer or interface that is in the center of the user and PC hardware. An OS is normally a software program, which performs fundamental tasks such as memory management, document management, managing I/O, process management and managing peripheral devices such as printers and disk drives. A well-known OS includes Windows, Linux, OS X, OS/400, VMS, AIX, z/OS, and many more. The following diagram depicts the different building blocks of an OS:

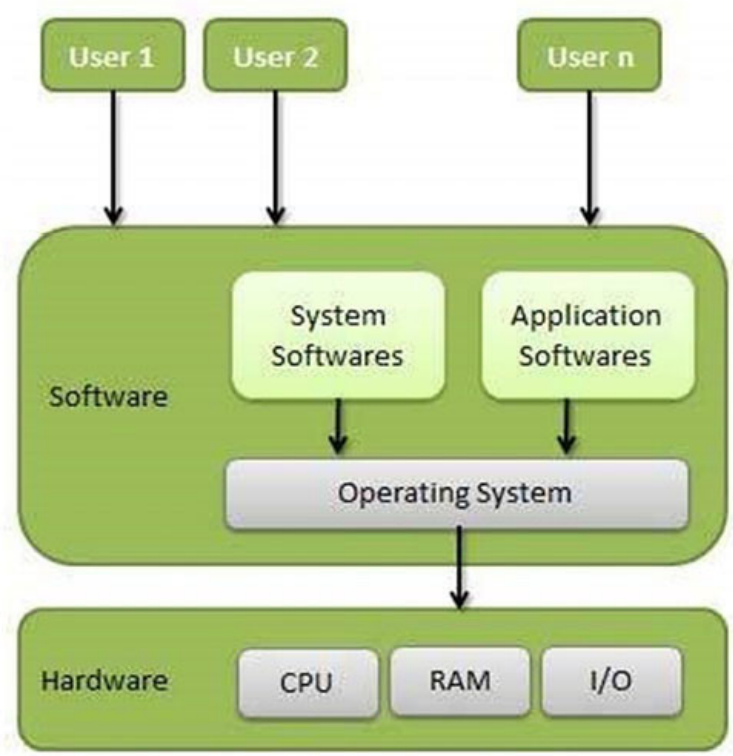


Figure 5.1: Conceptual view of an Operating System

An OS is a software application that functions as a mediator between the user and the hardware and handles the execution of different applications.

The following are the key capabilities of an OS:

Memory management

File management

Processor management

Security

Device management

Error detecting aids

Controlling system performance

Job accounting

Coordination between other software and users

Explain re-entrancy.

It is a memory storage saving methodology for multi-program timesharing systems. A re-entrant method is normally one where many users share one copy of an application during the same time. The re-entrancy provides two crucial aspects: program will remain unchanged, and the data for each of the process is maintained separately. Hence, the permanent section is the program, and the transient section is the pointer pointing to the local variables. A re-entrant procedure can be interrupted by an interrupting method and still have the capability to execute properly on returning to the caller. The following diagram shows the key capabilities of an OS:

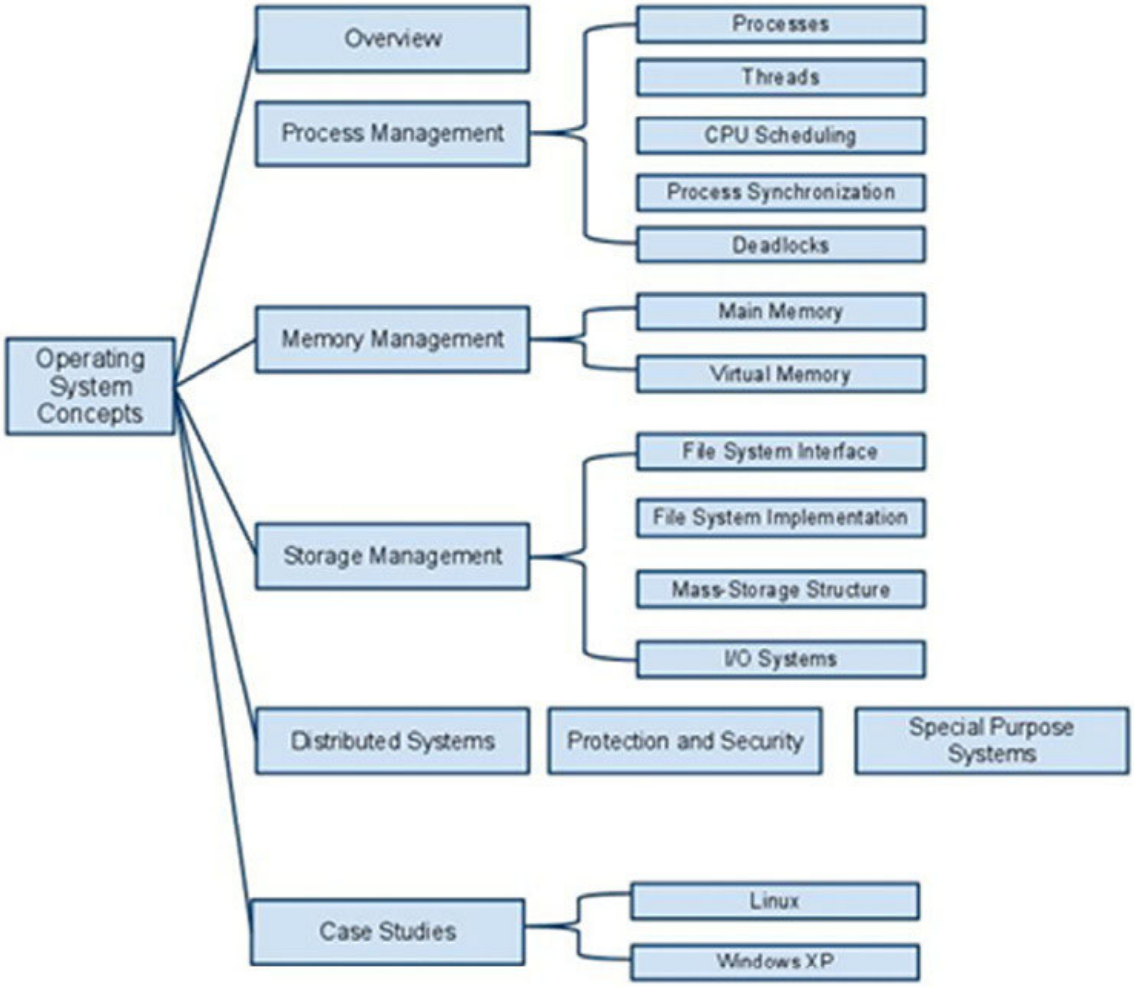


Figure 5.2: Key capabilities of an Operating System

Explain short term, medium term and long term scheduling.

Shortterm schedule: This is also called the dispatcher, which executes frequently and makes the vital decision on which the next process should execute. When an event occurs, the scheduler is usually invoked and results in the interruption of a process by pre-emption.

Mediumterm scheduler: This is a section of a swapping function and relates to processes that are in the blocked or suspended state. The processes are swapped into the memory until they are ready for execution. This swapping decision is done based on the memory management methodology.

Long term scheduler: This scheduler manages the processes that are admitted to the system for processing. This scheduler also manages multi-programming. Once admitted in the list, the work becomes a process activity.

What is the response and turnaround time?

Response time: The response time is defined as the time interval between submission of the request and the initial response from the process.

Turnaround time: The turnaround time is defined as the time interval between submission of a request and its completion.

Explain key thread-scheduling strategies.

Dedicated processor assignment: This plan facilitates implicit scheduling described by the allocation of threads to processors. The process is then allocated to processors totaling to the number of threads in the process. These processors are selected from the obtainable pool.

Load sharing: The global queue of threads is managed and processes are not assigned to a specific processor. Each idle processor selects a thread out of this queue. Load balancing is normally a technique wherein the tasks are assigned to processors on a long-term basis.

Dynamic scheduling: The number of program threads can be changed during execution.

Gang scheduling: A collection of correlated threads is scheduled to run on processors parallelly. Related processes are scheduled to reduce synchronization blocking and process switching.

How are signal and wait functions for a monitor different from that of semaphores?

Monitor: If a process in a monitor signals and there is absolutely no task, waiting upon this condition, the signal is lost and it facilitates a less difficult program design.

Semaphores: In semaphores, all functions and events have an impact on the value of the semaphore; therefore, the signal/wait functions need to be balancedout in the application program. A binary semaphore takes only 0 or 1 as values and are utilized to create mutual exclusion and synchronize concurrent processes.

What are the key layers of Windows?

Kernel

Hardware abstraction layer - HAL

Subsystems

System services

What is symmetric multi-processing?

Symmetric multiprocessing (SMP) is a mode of operation to accomplish reliability and optimum efficiency. In this technique, any processor threads can be assigned to any processors.

What is a deadlock?

A deadlock is a situation or condition where each of the two processes for each other to complete so that they can start. This results in both the processes to hang.

What is starvation and aging?

Starvation is a resource management problem where a process does not get the resources it needs for a long time because the resources are being allocated to other processes. Aging is a technique to avoid starvation in a scheduling system.

What are different types of scheduling algorithms?

There are four different types of scheduler:

First come first serve (FCFS): The first came process is served first.

Round Robin (RR): Each process is given a quantum amount of time.

Shortest job first (SJF): Process with the lowest execution time is given first preference.

Priority scheduling (PS): Priority value called (nice value) is used for selecting a process. Its value is from 0 to 99. 0 being the max and 99 being the least.

Why is the round robin algorithm considered better than the first come first served algorithm?

The first come first served algorithm is the simplest scheduling algorithm known. The processes are assigned to the CPU on the basis of their arrival time in the ready queue. Since it is non-preemptive once a process is assigned to the CPU, it will run until completion. Since a process takes the CPU until it is executed, it is not very good in providing good response times. It can make other important processes wait unnecessarily.

On the other hand, the round robin algorithm works on the concept of time slice or also known as quantum . In this algorithm, every process is given a predefined amount of time to complete the process. In case, a process is not completed in its predefined time, then it is assigned to the next process waiting in queue. In this way, a continuous execution of processes is maintained which would not have been possible in case of the FCFS algorithm.

What are the various IPC mechanisms?

The various Inter Process Communication (IPC) mechanisms are as follows:

Sockets

Pipes

Shared memory

Signals

Message queues

Process management

What are the elements of a process image?

User program: This is the program code that is executed when the application runs.

User data: This is actually the modifiable section of the user program, which includes the user stack, program data and programs, which may be modified.

Process control block (PCB): This is data required by the operating system kernel to manage a particular process.

System stack: Each process will have one or more stacks linked to it and it is utilized to store and manage local parameters. This is also utilized to invoke addresses of methods and make system calls.

What do you understand by process spawning?

When the operating system at a request from another process creates a new process, this mechanism is known as process spawning . Page replacements or swapping is also known as page cannibalizing.

What are the triggers for process termination and process suspension?

Process termination :

Normal completion

Memory not available

Bounds violation

Exceeded time limit

Arithmetic andprotection error

Input - Output failure

Operator or operating system intervention

Termination of parent

Privileged andinvalid instruction

Process Suspension:

Timing and swapping

Request parent process

Request interactive user

What is process migration?

Process migration is the act of transferringa process state from the source machine to the target machine.

What is a thread?

A thread is a program line under execution. A thread, sometimes called a lightweight process, is a basic unit of CPU utilization; it comprises a thread id, program counter, register set, and stack.

What is process synchronization?

A situation where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place is called the race condition. To guard against the race condition, we need to ensure that only one process at a time canmanipulate the same data. The technique we use for this is called process synchronization . The different synchronization mechanisms are as follows:

Semaphores

Mutex

Monitors

Condition variables

Critical regions

Read/Write locks

What is context switching?

Transferring the control from one process to the other process requires saving the state of the old process and loading the saved state for a new process. This task is known as context switching.

What is multitasking?

Multitasking is the process within an operating system that allows the user to run several applications at the same time. However, only one application is active at a time for user interaction although, some applications can run behind the scene .

What is pre-emptive multitasking?

Pre-emptive multitasking allows an operating system to switch between software programs. This, in turn, allows multiple programs to run without necessarily taking complete control over the processor and resulting in system crashes.

What is the advantage of a multiprocessor system?

As the processors are increased, you will get a considerable increment in the throughput. It is also cost effective because they can share resources. So, the overall reliability increases.

Which are the necessary conditions to achieve a deadlock?

There are four necessary conditions to achieve a deadlock:

Mutual exclusion : At least one resource must be held in a non-sharable mode. If any other process requests this resource, then that process must wait for the resource to be released.

Hold and wait: A process must be simultaneously holding at least one resource and waiting for at least one resource that is currently being held by some other process.

No pre-emption: Once a process is holding a resource (that is, once its request has been granted), then that resource cannot be taken away from that process until the process voluntarily releases it.

Circular wait: A set of processes {P0, P1, P2, . . ., PN} must exist such that every P[i] is waiting for P[(i + 1) % (N + 1)] .

What is spooling?

Spooling is a process in which data is temporarily gathered to be used and executed by a device, program or the system. It is associated with printing. When different applications send the output to the printer at the same time, spooling keeps all these jobs into a disk file and queues them accordingly to the printer.

Explain the basic functions of process management.

The basic functions of the OS process management are as follows:

Allocating resources to processes

Enabling processes to share and exchange information

Protecting the resources of each process from other processes

Enabling synchronization among processes

Difference between a process and a program

A program is a set of instructions for a computer to perform a designated task, whereas a process is an operation, which takes the given instructions and performs the manipulations as per the code called execution of instructions. A process is entirely dependent of a program .

A process is a module that executes modules concurrently. They are separate loadable modules; whereas a program performs the tasks directly relating to an operation of a user such as word processing, executing presentation software, and many more.

Memory and file management

What is thrashing?

This is a virtual memory management technique when the processor dedicates time and effort swapping pages, rather than executing the code and it is due to a lot of page faults.

Describe the buddy memory allocation system.

Free memory is normally managed using connected lists of structure; each of equivalent-sized blocks (size 2^k). When an application needs memory, the block size of the next higher order is selected and partitioned into two parts. Remember that both the sections will differ in the address within their kth bit and these parts are referred to as buddies. When any leveraged block is usually freed, the OS validates to find if its buddy can be free. When this happens, it has re-joined and placed into the original free block list.

In memory management what are placement and replacement algorithms?

Placement algorithms control wherein the available real memory space to load an application. Frequently leveraged strategies are best-match, next-fit, first-fit.

Replacement algorithms are utilized when memory space is running full, and one process section needs to be swapped out to accommodate another process section. The replacement algorithm decides which section needs to be swapped.

Compare runtime dynamic linking and load time dynamic linking

Runtime dynamic loading: In this scheme, the linking is usually postponed till the references are updated during execution and the right program module is normally loaded into the memory space.

Loadtime dynamic linking: In this scheme, a program module to be loaded is usually read into the memory at load time. Reference to a target external module will also cause the module to be loaded into the memory.

What do you understand by pre-paging and demand paging?

Pre-paging: Pages other than those demanded by page faults are swapped into the memory. The decision about pages is made basis the common access patterns.

Demand paging: A page is swapped into the memory only when a section on that page is referenced during the program execution.

What is virtual memory?

Virtual memory is a hardware technique where the system appears to have more memory that it actually does. This is done by timesharing, the physical memory and storage parts of the memory disks when they are not actively being used.

What is the meaning of physical memory and virtual memory?

Physical memory is the actual real memory used in RAM. Virtual memory as the name suggests is not real. The OS uses the virtual memory as a memory management technique in which the non-contiguous memory is presented to software as the contiguous memory. If the RAM falls short of memory to accommodate more running processes, the OS allocates a portion of your hard drive to act as though it were RAM. That's what is referred to as virtual memory.

What is fragmentation? Explain the different types of fragmentation

When many of the free blocks are too small to satisfy any requests, then fragmentation occurs. External fragmentation and internal fragmentation are two types of fragmentation. External fragmentation happens when a dynamic memory allocation algorithm allocates some memory and a small

piece is left over that cannot be effectively used. Internal fragmentation is the space wasted inside the allocated memory blocks because of restriction on the allowed sizes of allocated blocks.

What is cache memory?

Cache memory is random access memory (RAM) that a computer's microprocessor can access more quickly than it can access the regular RAM. As the microprocessor processes data, it looks first in the cache memory and if it finds the data there (from a previous reading of data), it does not have to do more time-consuming reading of data from the larger memory.

What is logical and physical addresses space?

Logical address space is generated from the CPU; it bounds to a separate physical address space, which is central to proper memory management. Physical address space is seen by the memory unit. Logical address space is the virtual address space. Both these address spaces will be the same at compile time but differ at the time of execution.

What is throughput, turnaround time, waiting time, and response time?

Throughput: Number of processes that complete their execution per time unit

Turnaround time: Amount of time required to execute a particular process

Waiting time: Amount of time a process has been waiting in the ready queue

Response time: Amount of time it takes from when a request was submitted until the first response is produced, not the output (for a time-sharing environment).

What is a MemoryManagement Unit (MMU)?

MMU is a hardware device that maps a virtual program address to a physical address (memory). As part of the MMU operations, the value in the MMU relocation register is added to every address generated by a user program at the time it is sent to memory. The user program only deals with logical addresses; it never sees the real physical addresses. MMU deals with conversation of physical to virtual memory and this process is transparent to the users This MMU mechanism provides a large virtual memory for the end user programs than provided through actual physical memory of the computer.

What is a daemon?

A daemon is a program that runs in the background without the users' interaction. A daemon runs in a multitasking operating system like UNIX. A daemon is initiated and controlled by special programs known as processes.

What is pre-emptive and non-preemptive scheduling?

Pre-emptive scheduling: The pre-emptive scheduling is prioritized. The highest priority process should always be the process that is currently utilized.

Non-preemptive scheduling: When a process enters the state of running, the state of that process is not deleted from the scheduler until it finishes its service time.

What are the disadvantages of context switching?

Time taken for switching from one process to other is a pure overhead. Because the system does not do any useful work while switching, so one of the solutions needs to go for threading whenever possible.

What is a data register and address register?

Data registers can be assigned to a variety of functions by the programmer. They can be used with any machine instruction that performs operations on data.

Address registers contain the main memory addresses of data and instructions or they contain a portion of the address that is used in the calculation of the complete addresses.

Multithreading and synchronization

When is the state of a system safe?

A set of processes are in a safe state when there is at least one temporal order where all processes can run till the completion without causing a

program deadlock.

What is an idle-thread?

The special thread known as a dispatcher will be executed when there is no ready thread/s.

What are the different states for a thread?

Ready

Standby

Running

Waiting

Transition

Terminated

What is time slice?

The timer in the CPU is set to interrupt every N milli second where this N is called the time slice.

It is the time each user gets to execute before the control is given to the next user.

At the end of each time slice, the value of N is incremented and the record is maintained.

It also maintains the record of the total time the user program has executed so far.

This method helps in time sharing among the various users.

Miscellaneous

What are Coffman's conditions that may lead to a deadlock?

Hold and wait: In this condition, a process can be allocated a few resources while waiting for the other resources.

Mutual exclusion: In this condition, one process may utilize a critical resource at one time.

Circular wait: In this condition, there is closed chain of processes where each process holds at least one resource required by another process in the chain.

No pre-emption: In this condition, any resource can be forcibly removed from a process holding it.

What is cycle stealing?

Cycle stealing is encountered regarding Direct Memory Access (DMA). Cycle stealing is a mechanism when the DMA controller leverages the data bus when the CPU does not need it or it can enforce the CPU into temporarily suspension. Cycle stealing is usually accomplished at definite break points within a CPU instruction cycle.

What is arm stickiness?

This occurs in scenarios when one or more processes have a higher access rate to data using one track of a storage disk, and essentially monopolizing the device by frequent requests on that same track. This occurs with the most common device scheduling algorithms. High-density, multi-surface storage disks are likely to be affected by arm stickiness.

What is meant by busy waiting?

During this period, the central processing unit is not involved in any productive activity and the process does not progress any further. The repeated execution of the program loop while waiting for an event to occur is known as busy-wait .

What is a trap and trapdoor?

The trap door is a top secret undocumented entrypoint into a program and is utilized to grant access without normal ways of access authentication. The trap is normally a software program interruption and is a result of an exception or error condition in this program.

Define latency, transfer and seek time with regards to disk input/output.

Access time: Summation of latency and seek time is the access time.

Seek time: Seek time is the time required to move the disk arm to the required track.

Transfer time: Time taken to transfer the data set is usually the transfer time.

Rotational latency: Rotational latency or delay is the time taken to reach the head from the beginning of the requested sector.

What is a command interpreter?

The part of an OS that interprets commands and carries them out is a command interpreter.

A command interpreter is the part of a computer operating system that understands and executes commands that are entered interactively by a human being or from a program. In some operating systems, the command interpreter is called the shell.

The BIOS looks for the files needed to load in case of Windows, which are the Command.com . The required files are Command.com , IO.sys, and Msdos.sys to get Windows started. They reside in the root of the C drive.

Conclusion

An operating system controls and coordinates the use of the hardware among the various applications programs for various uses. An operating system acts as a resource allocator and manager. The operating system is a control program, which controls the user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of I/O devices.

This chapter covered the question bank for the operating system domain. This included core concepts, process management, memory and file management, multi-threading, and synchronization. The next chapter will cover the object-oriented programming domain, which will include arrays, collections, structures, interfaces, classes, and inheritance.

CHAPTER 6

Object-oriented Programming (OOPs)

Object-oriented programming also known as OOP is a paradigm which is usually cantered around the concept of classes and objects and which includes data, known as attributes; and functions, known as methods. In the OOP paradigm, the object's methods can access and change the data fields. In the OOP paradigm, applications are designed by creating objects that interact with each other. There is usually a considerable variety of OOP languages, but many widespread ones are structured around the class, that is, objects are instances of classes, which also govern their type.

This chapter covers the OOPs aspects and provides the question bank which includes topics for core concepts, encapsulation, composition and inheritance, and polymorphism.

Concepts

What is object-oriented programming (OOPs)? What are classes and objects?

Object-oriented programming (OOPs) is a model to create logical modules known as classes consisting of properties, fields, methods, and events. An object is usually created by the program from a class definition. This object encapsulates the features such as behavior and data. OOP facilitates developing modular applications and assembles them as software programs.

Class: A class explains the attributes and functions that define the behavior. A class is a template for an object. A class is a complete data type, which represents a template for the objects. A class is the fundamental building block of the OOPs application. This is usually a template that defines the properties, behaviors, events for a particular group of objects. The class consists of behavior and data. For example, the aircraft class contains data such as category, model number, color, and behavior such as length of the flight time, speed, and total number of passengers. A class can inherit the behaviors and data from the other classes by extending these classes, which are known as base or parent classes.

Object: An object is an instance of a class and is a fundamental building block of the OO program. An object is usually an entity consisting of behavior and attributes. Attributes and behavior for an object are described in the class definition.

What is the relation between an object and a class?

A class acts as a blueprint or a template that defines the behaviors, states and properties and that are common to objects. Objects are instances of a class. For example, in a class, Person, Math Teacher and Footballer will be derived classes. One can create n number of objects for the class Person such as Math Teacher and Footballer . To create an object of a class, the new operator is utilized. When an object is usually instantiated, the application allocates memory for the entire data members that are present in that particular class.

Explain the concepts of OOPs.

The following are the key concepts of OOP:

Polymorphism: This facilitates leveraging an entity in various forms.

Abstraction: This is a method of publishing just the relevant and important data to the end-users without exposing unnecessary details to the outside world.

Inheritance: This promotes code reusability and eliminates the use of unnecessary programs. This feature of OOP enables the child class to acquire all the features from its parent class.

Encapsulation: This prevents undesirable access to data by the packaging code and data in a one unit known as an object.

Procedural programming is normally structured around a modular approach where- huge programs are divided into multiple procedures. A procedure is normally a group of instructions that are executed in a sequential manner, whereas OOP is usually centered on objects. An object contains several aspects, including variables and methods.

In the procedural programming paradigm, there are not any access specifiers, which means the class data members will be accessed from anywhere in the application. One can define the scope of a specific data member by leveraging access specifiers such as private, protected, public in the OOPs paradigm.

The prevalent programming languages, for example, C++ and Java are multi-paradigm programming languages, which support object-orientation in combination with procedural programming. Most common OO languages are C++, C#, Java, Python, Ruby, PHP, Objective-C, Perl, Scala, and Smalltalk. The following diagram depicts the various pillars of OOPs:

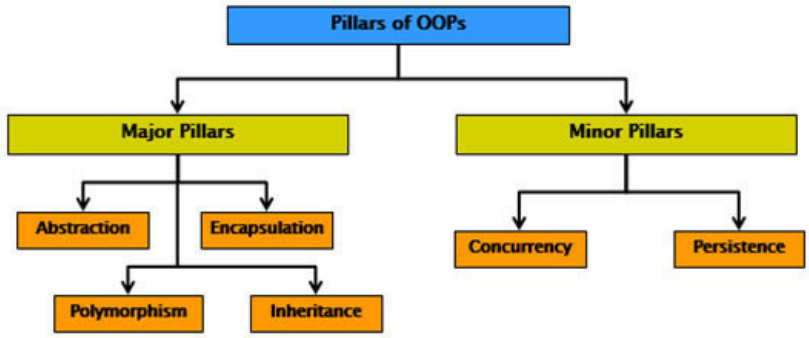


Figure 6.1: Core Pillars of OOP

What are the key benefits and characteristics of OOPs?

The main benefits of the OOPS concepts are modularity, extensibility, simplicity, reusability, maintainability, modifiability, and many more. The complexity of the programming can be reduced and the coding structure can be made clear. The different complex functionalities can be decoupled using different classes and implementation methods around the application. The reusability feature provides minor changes in the code whenever needed, which provides the adaptability for the code changes or functionality changes.

The objects of the different classes can be reused in different implementation classes to use its features completely. The maintenance process becomes easier if the code is maintained in an organized way.

The benefits of OOPSs are as follows:

Real-world programming

Reusability of code

Modularity of code

Resilience to change

Information hiding

What is the difference and similarity between a structure and class?

Class:

A class is a reference data type.

Classes support inheritance.

Once the class is instantiated, runtime allocates memory for the instance in a heap.

The class contains a constructor and destructor.

A null value can be assigned to variables of a class.

Structure:

A structure is a value type.

Structures do not provide inheritance.

Memory is allocated on the stack.

A null value cannot be assigned to a structure.

Constructors and destructors are present in structures.

Runtime automatically initializes structures.

Structures and classes are the two most critical and fundamental building blocks that can be leveraged to create modular applications by leveraging OO concepts.

Comparing classes and structures:

The access ranks for classes and struct members, including structs and nested classes, are by default private.

To restrict the access of the data members and methods, access specifiers such as public, private and protected are utilized in classes and structures.

Classes and structures both will have constructors, destructors, methods, properties, constants, enumerations, events, and delegates.

Both classes and structures can extend interfaces to utilize the OO concept, namely, multiple-inheritance.

What do you understand by a delegate and multicast delegate?

A delegate is a class leveraged for storing and managing a reference to a method and invoking this method at runtime. A delegate stores and manages reference to just those methods whose signatures is a match with that of the delegate.

In practice, it is feasible for a delegate to hold references to multiple methods and such delegates are known multicast delegates.

Explain the concept of the constructor, destructor, and static constructor.

Constructor: When the instance of a class is being constructed the runtime invokes the constructor automatically, which is a special class method. The constructor is utilized to initialize the class data members. The constructor is declared and defined with the same name as the class name. The key features of constructors are as follows:

Constructors do not return any parameters.

In a given class, one can have overloaded constructors.

Declaring a constructor is not mandatory.

Destructor: When the instance of a class is being destroyed, the runtime invokes the destructor automatically, which is a special class method. A destructor is leveraged to free the allocated memory, which also releases the resources. One can write a custom class method that facilitates controlling the object destruction by calling it from the destructor. The destructor name is the same as that of the class but has a prefix tilde (~). The key characteristics of the destructor are as follows:

One cannot overloaded destructors.

Destructors do not return any parameters.

Destructors are also always public similar to constructors.

Static constructor: Static constructors are utilized to initialize the static class data members. The runtime invokes the static constructor before creating the first instance of the class. The key features of the static constructor are as follows:

Only one static constructor can be defined in the class.

A static constructor does not need access modifiers.

A static constructor does not accept any parameters.

A static constructor is invoked just once when the program begins the execution.

A static constructor will only be able to access static class members.

What is an abstract class and its characteristics?

An OOPs concept where the abstract class is only leveraged as a base class but which cannot be instantiated in the user program. The key features of an abstract class are as follows:

One cannot instantiate an abstract class and therefore, it can be only leveraged as the base class.

There should be at least one abstract method declared in the abstract class.

The access specifier for the abstract class is always public.

The abstract class is declared leveraging the keyword abstract.

One may define abstract as well as non-abstract members in the abstract class.

The OOP concept of inheritance would be able to leverage via an abstract class, which facilitates the common definition in a base class. The following diagram depicts the rules for an abstract class:

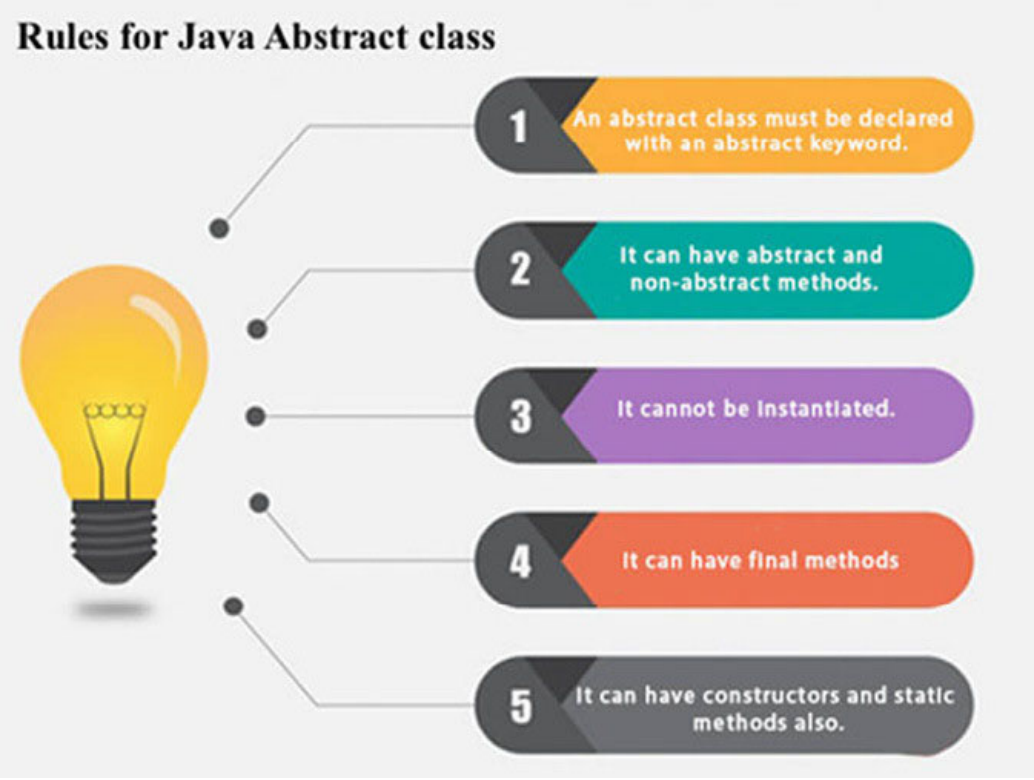


Figure 6.2: Characteristics of an abstract class

Differentiate between an abstract class and interface.

Abstract class:

A class can only extend one abstract class.

Members of the abstract class can be protected or private.

One can define methods of an abstract class as abstract or concrete.

An abstract class can be extended by any other class.

Derivedclasses may or may not implement all its methods.

An abstract class can have a constructor defined.

The method of implementation can be provided by an abstract class.

Interface:

A class can implement multiple interfaces.

Classes should provide implementations for interfaces'definitions.

A class implementing the interface must implement all the methods of the interface.

An interface only has public members.

All methods of an interface must be declared abstract.

Interfaces do not have any constructors.

Only an interface can extend another interface.

There is no definition for any of its method in interfaces.

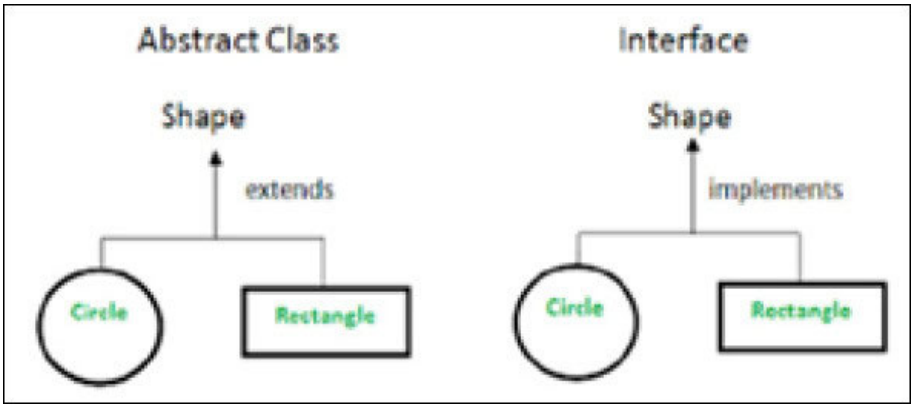


Figure 6.3: Abstract class vs interface

What are stacks and queues?

Stacks are data structures wherein all data items are accessed and processed basis the LIFO technique. In a stack, elements are inserted and deleted from only one end called the top of the stack.

The queue is a data structure in which insertion and deletion of a data item are done on the FIFO technique. Data items are inserted from the one end, the rear end, and are deleted from the additional end, the front end of the queue.

What is a structure?

The structure is a heterogeneous collection of data items referenced by one name and the declared utilizing the keyword struct . The following program code creates a strut to manage an employee's details:

```
struct emp

{

  fixed int employeeID[15];
```

```
  fixed char employeename[30];

  fixed char employeedept[15];

  fixed char employeeaddr[50];

  fixed char employeedesig[15];

}
```

What is an inline function?

An inline function is a technique used by the compilers and instructs to insert the complete body of the function wherever that function is used in the program source code.

What is a friend function?

A friend function is a friend of a class that is allowed to access to public, private, or protected data in that same class. If the function is defined outside the class, it cannot access such information.

A friend function can be declared anywhere in the class declaration, and it cannot be affected by access control keywords like private, public or protected.

What is function overloading?

Function overloading is a normal function, but it can perform different tasks. It allows the creation of several methods with the same name which differ from each other by the type of input and output of the function.

Example:

```
void add(int& a, int& b);

void add(double& a, double& b);

void add(struct bob& a, struct bob& b);
```

The following diagram depicts the function overloading mechanism in OOPs:

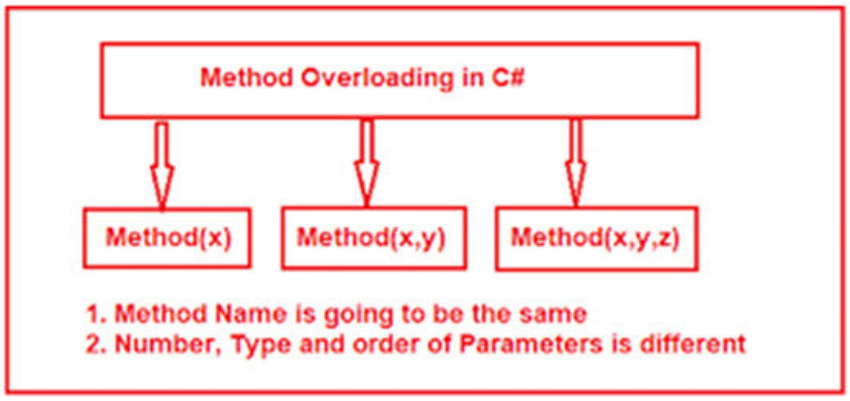


Figure 6.4: Function overloading

What is operator overloading?

Operator overloading is a function where different operators are applied and they depend on the arguments. Operators such as +, -, * can be used to pass through the function, and they have their own precedence to execute.

What is a ternary operator?

The ternary operator is said to be an operator which takes three arguments. Arguments and results are of different data types, and it depends on the function. The ternary operator is also called a conditional operator.

What is the use of the finalize method?

The finalize method helps to perform cleanup operations on the resources, which are not currently used. The finalize method is protected, and it is accessible only through this class or by a derived class.

What are the different types of arguments?

A parameter is a variable used during the declaration of the function or subroutine and arguments are passed to the functions, and they should match with the parameters defined. There are two types of arguments.

Call by value: The value passed will be modified only inside the function, and it returns the same value whatever it is passed to it to the function.

Call by reference: The value passed will be modified in both inside and outside the functions and it returns the same or different value.

The following diagram depicts call by value vs call by reference models in programming:

Figure 6.5: Call by value vs call by reference

What are tokens?

A token is recognized by a compiler, and it cannot be broken down into component elements. Keywords, identifiers, constants, string literals, and operators are examples of tokens. Even punctuation marks are also considered as tokens such as brackets, commas, braces, and parentheses.

How can we call the base method without creating an instance?

Yes, it is possible to call the base method without creating an instance. And that method is the static method. Doing inheritance from that class, use the base keyword from a derived class.

What are the various types of constructors?

There are three types of constructors, which are as follows:

Default constructor: This has no parameters.

Parametric constructor: With parameters. It creates a new instance of a class and passes arguments simultaneously.

Copy constructor: This creates a new object as a copy of an existing object.

What is early (static) and late (dynamic) binding?

Early binding refers to the assignment of values to variables during design time, whereas late binding refers to the assignment of values to variables during runtime. The following diagram depicts the difference between stack vs dynamic polymorphism and the methods to achieve these in the OOPs:

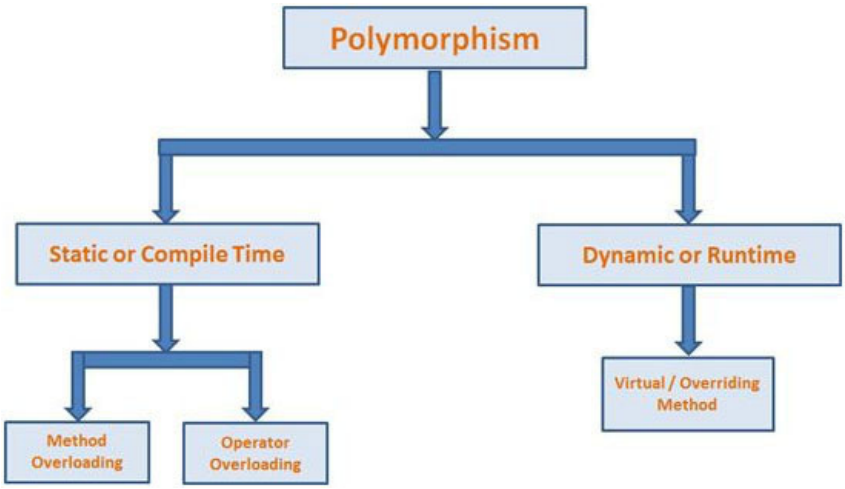


Figure 6.6: Family of data structures

What is ‘this’ pointer?

This pointer refers to the current object of a class. The this keyword is used as a pointer which differentiates between the current object with the global object. Basically, it refers to the current object.

What is the difference between a structure and a class?

In a structure the default access type is public, but in the class the default access type is private. A structure is used for grouping data whereas a class can be used for grouping data and methods. Structures are exclusively used for data, and they do not require strict validation, but classes are used to encapsulates and inherit data which requires strict validation rules.

What are access modifiers?

Access modifiers determine the scope of the method or variables that can be accessed from other various objects or classes. There are five types of access modifiers, which are as follows:

Private

Protected

Public

Friend

Protected Friend

Encapsulation

Describe the characteristics of an interface.

An interface is a blueprint that consists of method signatures also known as method declarations. The method signature consists of numbers of attributes, types of attribute values, references or return types. An interface provides no implementation of methods because it contains just the declarations of methods with no method body. One cannot instantiate an interface.

The features of an interface are as follows:

An interface is utilized to implement multiple inheritances. This is the key difference from the abstract classes as a class can't inherit from more than one class but can inherit multiple interfaces.

An interface declares and defines specific methods with the attributes.

Methods in the interface must be public and abstract but the data members are to be declared as public, static and final.

All the methods of the interface must be implemented by aclass implementing an interface.

What do you understand by encapsulation and abstraction?

Encapsulation: Data encapsulation is a key OO concept that facilitates binding code and data in a single translation unit called a class and hiding the implementation details from outside world. This prevents un-warranted access to data members and restricts the enduser to leverage the public methods of the class. Encapsulation implements one of the four fundamental OOP concepts, that is, data hiding. An object binds together the properties and methods that manipulate the data, which is safe from outside manipulations. Encapsulation is usually the process of hiding the elements from the outside world of an object that do not contribute to its key features.

Abstraction: Abstraction is the mechanism of hiding unwanted details of an object from the outside world, and exposing just relevant details. The best example of abstraction is the interface. Encapsulation hides the unimportant details of an object from the outside world and abstraction makes just the relevant information of an object available to the outside world.

The following table compares abstraction and encapsulation:

Abstraction	Encapsulation
Abstraction is a general concept formed by extracting common features from specific examples or The act of withdrawing or removing something **unnecessary**.	Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps both **safe from outside interference** and **misuse**.
You can use abstraction using **Interface** and **Abstract** Class	You can implement encapsulation using **Access Modifiers** (Public, Protected & Private)
Abstraction solves the problem in **Design** Level	Encapsulation solves the problem in **Implementation** Level
For simplicity, abstraction means hiding implementation using Abstract class and Interface	For simplicity, encapsulation means hiding data using getters and setters

Figure 6.7: Abstraction vs Encapsulation

Composition and inheritance

What are the types of inheritance?

Inheritance is an OO feature of a class to leverage the properties and methods of another class while adding its own features. It enables you to add fresh features and functionalities to an existing class without changing the existing class. There are four types of inheritance in OOPs:

Single inheritance: This mechanism consists of one derived class and one base class. The following diagram depicts a single inheritance:

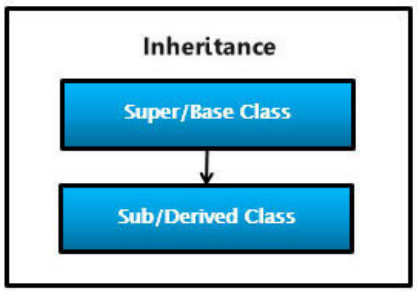


Figure 6.8: Single inheritance

Hierarchical inheritance: This mechanism consists of one base class and multiple derived classes from this base class.

Multi-level inheritance: This mechanism consists of a class derived from a derived class.

Multiple-inheritance: This mechanism consists of several base classes and one derived class. The following diagram depicts a multiple inheritance:

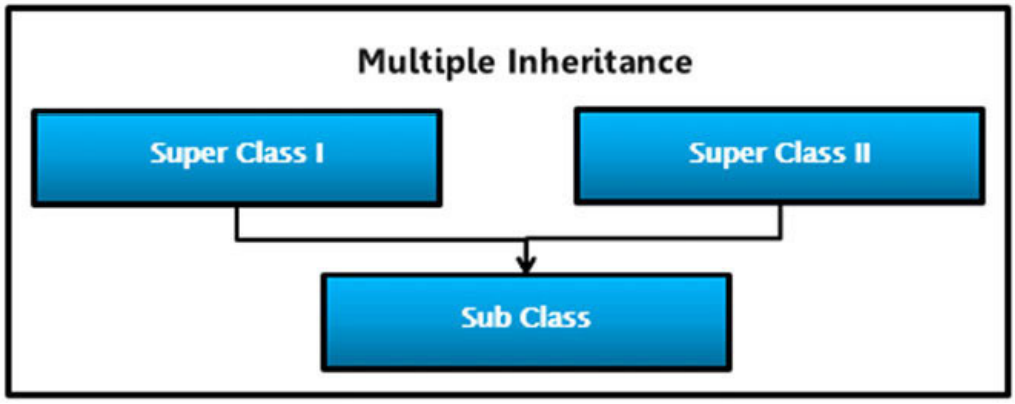


Figure 6.9: Multiple inheritance

What is a subclass and superclass?

A subclass or derived class inherits data and methods from a parent or superclass.

A superclass or base class is the one from which other classes inherit data and methods.

By defining it with a keyword, we can restrict a class from being inherited. One cannot stipulate any other access specifier as all the methods in an interface are public, by default.

Can a parent class constructor be inherited?

No, a parent class constructor cannot inherit.

Differentiate between method overriding and method overloading?

Overriding is an OO concept that mandates the creation of two or more methods with the same name and same signature, that is, one in the parent class and the other in the derived class.

Overloading is an OO concept that facilitates leveraging a method with the same method name and different method signatures in the same class.

What is the super keyword?

The super keyword is used to invoke the overridden method, which overrides one of its superclass methods. This keyword allows you to access overridden methods and access hidden members of the superclass. It also forwards a call from a constructor to a constructor in the superclass.

What is the difference between aggregation and composition?

Aggregation and composition are the types of Association. In both aggregation and composition object of one class "owns" object of another class. But there is a subtle difference: Aggregation implies a relationship where the child can exist independently of the parent. Composition implies a relationship where the child cannot exist independent of the parent.

Composition is a restricted form of aggregation in which two entities are highly dependent on each other. In composition, both the entities are dependent on each other. When there is a composition between two entities, the composed object cannot exist without the other entity.

In the following diagram, a car and carburetor represents composition and a pond and duck represents aggregation. The following diagram depicts this composition vs aggregation relationship:

Figure 6.10: Composition vs Aggregation

What is the diamond problem in inheritance?

In multiple inheritance, suppose class A has two subclasses B and C and a class D has two super classes B and C . If a method present in A is overridden by both B and C but not by D , then from which class D will inherit that method B or C ? This problem is known as diamond problem. The following diagram depicts the diamond problem:

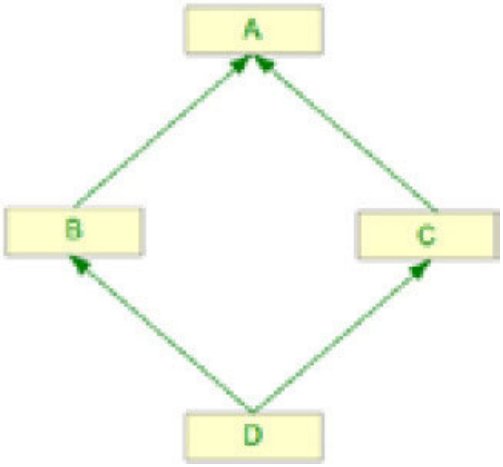


Figure 6.11: Classic diamond problem

Why Java does not support multiple inheritance?

Java was designed to be a simple language and multiple inheritance introduces complexities like the diamond problem. Inheriting states or behaviors from two different types of classes is a case, which in reality is very rare and it can be achieved easily through an object association.

Polymorphism

What is polymorphism?

Polymorphism is the feature of OOPs wherein an object can take multiple forms based on its type or class. The best example of polymorphism is invoking derived class functions through a base class reference at runtime. There are two types of polymorphism:

Static polymorphism also known as compile time polymorphism: Polymorphism that is resolved during the compiler time is known as static polymorphism. Method overloading is an example of compile time polymorphism. Method Overloading allows you to have more than one method having the same name, if the parameters of methods are different in number, sequence and data types of parameters.

Dynamic polymorphism also known as runtime polymorphism: Dynamic or runtime polymorphism is also known as method overriding in which a call to an overridden function is resolved during runtime, not at the compile time. It means having two or more methods with the same name, same signature but with different implementation.

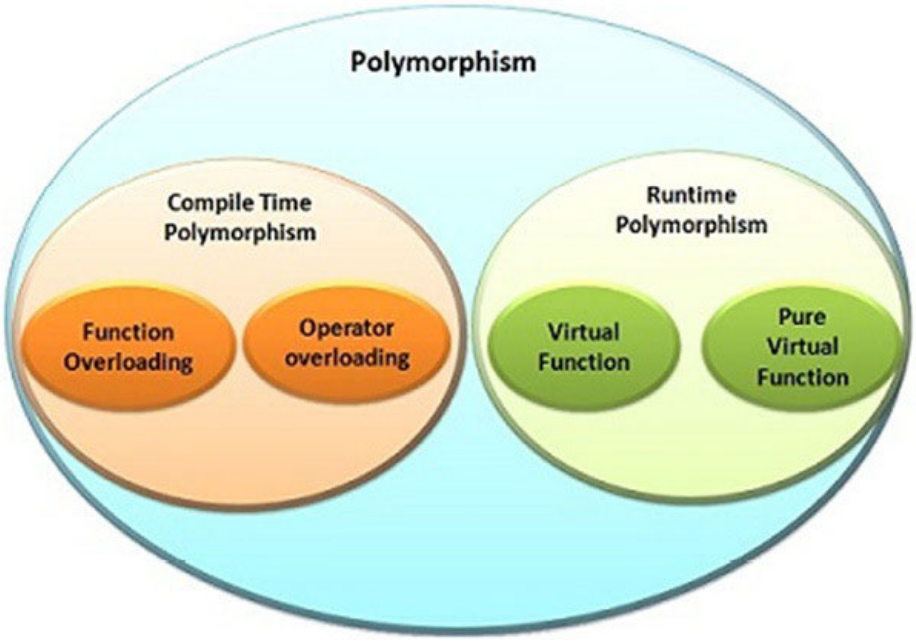


Figure 6.12: Different types of polymorphism

Describe method overloading.

Overloading is a OO concept that facilities declaring and defining several methods with the same name, but with different method signatures. Methods are identified based on:

Number of parameters

Types of parameters

Order of parameters

Example:

int addition(int a,int b) { return a+b;}

int addition (int a,int b,int c) { return a+b+c;}

float addition (float a, float b, float c) { return a+b+c;}

In case the original method is non-static can we declare an overridden method to be static?

The two virtual methods in the base and derived class must have the same method signatures.

What is the significance of the virtual keyword ?

A virtual keyword is leveraged while defining a class to mandate that the methods of the base class need to be overridden in the derived classes.

What is a virtual function?

A virtual function is a member function of a class, and its functionality can be overridden in its derived class. This function can be implemented using a keyword called virtual, and it can be given during function declaration. A virtual function can A token in C++, and it can be achieved in C Language using function pointers or pointers to function.

Can you prevent a method from being overridden but allow a class inheritance?

Yes, we should declare the class as public and declare the method as sealed.

Is it feasible to inherit private members of a class?

Private class members cannot be inherited as private members are available only to that class and not the outside world.

What is a pure virtual function?

A pure virtual function is a function which can be overridden in the derived class but cannot be defined. A virtual function can be declared as pure using the operator = 0 .

Virtual void function1() // Virtual, Not pure

Virtual void function2() = 0 //Pure virtual

Miscellaneous

Define enumeration.

Enumeration is a value data type that includes a collection of named values. These values are constants and known as enumerators. An enumeration is usually defined utilizing the keyword enum . Each enumerator is connected with the underlying type, by default, on the enumerator.

enum Fruits {pineapple, banana, peach, watermelon};

In the preceding example, enumeration Fruits is constructed, where number 0 is assigned to pineapple , number 1 to banana , number 2 to peach , and number 3 to watermelon . One can gain access to these enumerators through these values.

Is it a best practice to handle exceptions in code?

The application design must cater to handling exceptions in the application to deal with unexpected/scenarios that occur while the application is executing. For example, dividing a number by zero or passing a string value to an integer parameter and both these scenarios result in an exception.

try…catch…finally block: The try block embeds the programcode/sequence of instructions that may result in an exception and the catch block handles these exceptions. The catch block consists of the programcode that has to be executed when the exception condition occurs in the application.

To perform resource cleanups, the finally block is leveraged. If an exception occurs in the main try block, the program control transfers to the catch block and later to the finally block. If there is no exception, then the program control is still transferred to the finally block. Irrespective of the fact whether or not an exception is generated, the finally block will always execute.

To re-throw an exception in the application, the throws keyword is leveraged. This keyword signals the occurrence of an exception during an application run. When the program encounters athrowstatement , the function terminates and returns the error to the calling function.

Define an array?

An array is a collection of data elements stored in contiguous memory locations and referred by a name. The index of the array accesses different elements within an array. An index value of the array specifies the position of a particular data item in the array.

What are methods?

Methods are bundled together to share and process class data methods and they are fundamental building blocks in the OO paradigm. The method represents the behavior of a class and contains a sequence of program code statements. When declaring a method, one needs to specify the access specifier (public, private, and protected), method name, return value, and parameters.

What is a namespace?

A namespace is a container consisting of entities that have related functionality, which includes the classes.

Define an event.

The class generates a notification for other classes in the program whenever an action happens in the class and these notifications are called events. For example, when a mouse is clicked, the class generates an event called click. An event is declared leveraging the event keyword.

What is the difference between procedural programming and OOPS?

A procedural language is based on functions but an object-oriented language is based on real-world objects.

A procedural language gives importance to the sequence of function execution but an object-oriented language gives importance to states and behaviors of the objects.

A procedural language exposes the data to the entire program but an object-oriented language encapsulates the data.

A procedural language follows the top-down programming paradigm but an object-oriented language follows the bottom-up programming paradigm.

A procedural language is complex in nature so it is difficult to modify, extend and maintain but an object-oriented language is less complex in nature so it is easier to modify, extend and maintain.

A procedural language provides less scope of code reuse but an object-oriented language provides more scope of code reuse.

What is the meaning of 'IS-A' and 'HAS-A' relationship?

'IS-A' relationship implies inheritance. A subclass object is said to have 'IS-A' relationship with the super class or interface. If class A extends B, then A'IS-A' B. It is transitive, that is, if class A extends B and class B extends C, then A'IS-A'C. The instanceofan operator in Java determines the 'IS-A' relationship. The following diagram depicts the IS-A vs HAS-A relationship:

Figure 6.13: Is-A Vs Has-A relationship

What is dependency?

When one class depends on another because it uses that at some point in time, then this relationship is known as dependency . One class depends on another if the independent class is a parameter variable or local variable of a method of the dependent class. A dependency is drawn as a dotted line from the dependent class to the independent class with an open arrowhead pointing to the independent class. The following diagram depicts the different dependencies and associations:

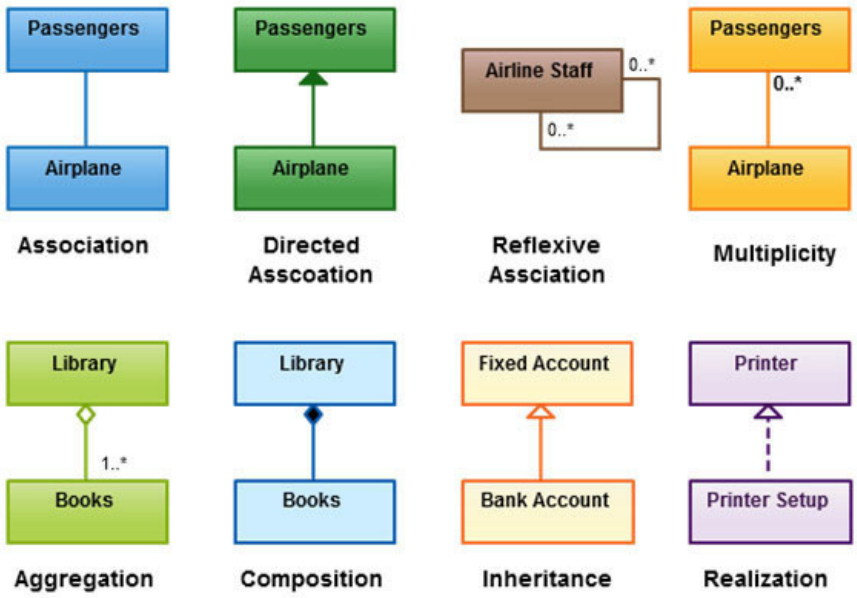


Figure 6.14: Various types of associations and dependencies

What is the difference between association and dependency?

The main difference between association and dependency is that in association, one class has an attribute or member variable of the other class

type but in case of the dependency, a method takes an argument of the other class type or a method has a local variable of the other class type.

What is method hiding in Java?

When you declare two static methods with the same name and signature in both superclass and subclass, then they hide each other. That is, a call to the method in the subclass will call the static method declared in that class and a call to the same method in superclass is resolved to the static method declared in the superclass.

Is Java a pure object-oriented language?

Java is not a pure object-oriented programming language. For example, there are many things you can do without objects, for example, static methods. Also, primitive variables are not objects in Java.

Conclusion

OOP is a technique to develop logical modules such as classes that contain properties, methods, fields, and events. An object is created in the program to represent a class. Therefore, an object encapsulates all the features such as data and behavior that are associated to a class. OOP allows developers to develop modular programs and assemble them as software. Objects are used to access data and behaviors of different software modules such as classes, namespaces, and sharable assemblies. The NET framework supports only OOP languages such as Visual Basic .NET, Visual C#, and Visual C++.

This chapter covers the OOPs aspects and provides the question bank for core concepts, encapsulation, composition and inheritance and polymorphism. The next chapter will provide a question bank for C/C++/Java programming languages which will include C/C++/Java, Pointers, memories, variables, data abstraction, inline functions, iterators, classes, exceptions, and serialization.

CHAPTER 7

C and C++ Programming

Aprogramming language consists of set of instructions that can be customized to yield various business outcomes. Programming languages consist of set instructions for a machine. Programming languages are utilized to create applications that solve specific domain problems.

A number of programming languages already exists, and many more are created each year. The programming language is described based on two key aspects one syntax (form) and second is semantics (meaning). A few languages are defined by specification artifactssuch asthe C programming language is specified by ISO while other languages such as Perl have a dominant implementation that is treated as a reference. Some languages will have both, the basic language reference and extensions from dominant implementation.

This chapter will cover the C and C++ Programming languages in detail. The section on C will include a question bank for aspects pertaining to preprocessing, functions, structures, pointer and memory management. The section on C++ includes a question bank for aspects pertaining to core concepts, keywords and operators, constructors and destructors, inheritance and polymorphism.

C programming

Preprocessing

What do you understand by #include and #include "file"?

This #include is leveraged to include a file. The difference between these mechanisms of file inclusion is in the sequence in which the preprocessor searches for these include files. When the pre-processor comes across the #include statement, it searches for the specified file in the angled brackets in the default location, that is, the path defined in the include environment variable. When the preprocessor encounters the #include “file” statement, it searches for the file in the current directory initially, and if it's not found in the current directory, then it looks for it in the default location, that is, the path defined in the include environment variable. #include handles all file formats like * .h . Regardless of the type of file, the preprocessor will include any file like mytest.z.

As shown in the following diagram, module1.c and module2.c leverage the sharedFunc() function that is defined in SharedImpl.c and declared in shared.h. To link and use the function, both module1.c and module2.c will have to include shared.h at the beginning of the files:

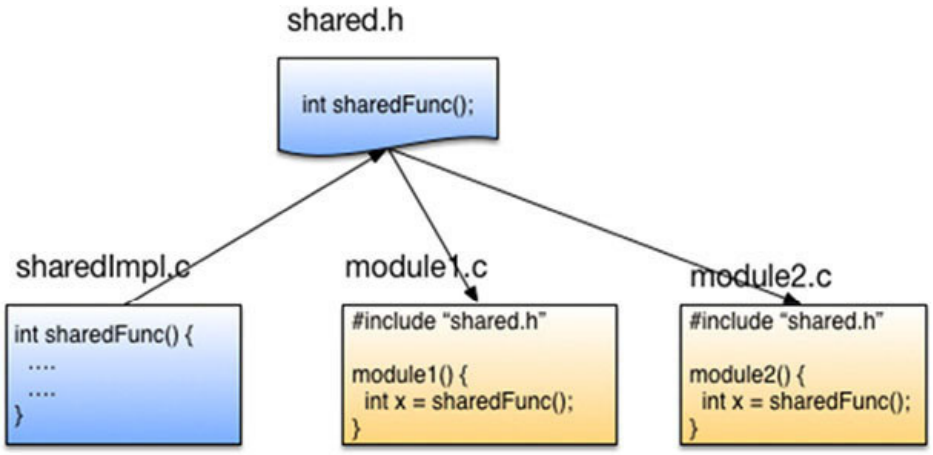


Figure 7.1: #include preprocessor directive

Can #include handle other file formats other than .h?

Yes # include hands other file formats. Regardless of the type of a file, the preprocessor or #include will include any file, for example, test.z .

How to restrict a header file from including more than once?

In C, to restrict multiple inclusions of a header file, one can leverage an include guard aka a macro guard. It is #ifndef - #endif. #ifndef maps to if not defined . For example:

ifndef GROUP_H

define GROUP_H

struct Example

{

int groupMember;

};

endif /* GROUP_H */

What are the standard predefined macros?

PI finds the ANSI C predefined macro:

LINE: Current line number

FILE: Name of the current file

TIME: Current time of compilation

DATE: Current date of compilation

__cplusplus: If the program is compiled in the C++ compiler

STDC: If ANSI C is followed by the compiler strictly

What is a pragma?

The #pragma preprocessor allows the compiler to exclude or include compiler-specific features. For example, if there is a feature xxx_yyy then,

pragma xxx_yyy(on)

forces the compiler to include the feature. Conversely, you can turn it off with the following line of code:

pragma xxx_yyy(off)

What is the benefit of using const over #define macro?

The difference is that a const variable is a real variable, which has a datatype and exists at runtime, and it cannot be altered later.

The #define macro is not a real variable, but it carries a constant value which replaces all the occurrences of that macro at the time of preprocessing.

Function and function pointers

What will be the output of the following code snippet?

float number1 = 6 / 4;

float number2 = 6 / 4.0;

printf("6/4 == %f or %f\n", number1, number2);

This is a case of the operator promotion:

The variable number1 is set to 6/4 . Because both 3 and 4 are integers, so the integer division is performed on them, and the result will be the integer value 1 .

The variable number2 is set to 6/4.0 . Because 4.0 is a float, the number 6 is converted to a float, and the result will be the floating value 1.5 .

Explain recursive functions and its advantages and disadvantages.

A recursive function is a function that calls itself.

The following are the advantages:

Recursive functions reduce the code size for applications like Tower of Hanoi.

Unwanted calling of functions can be avoided.

The following are the disadvantages:

The exit point must be created explicitly in the program, otherwise the stack overflow will take place

It is challenging to track the logic of the recursive functions, as they are complex.

Recursive functions are also difficult to debug.

Can the sizeof operator get the size of an array passed to a function call?

As it is a pointer to the data type of the array, the sizeof operator will not return the size of an array.

How can I convert a number to a string?

The following functions are utilized to convert integers to strings:

One can write custom functions too, as these functions are not safe, and they do not check whether the value passed to them is null or not.

What is a static function?

A static function is a special function whose scope is limited to the source file where the function is defined and cannot be leveraged outside this scope. This feature facilitates hiding some functions providing some standard interfaces or wrappers over that local function.

Explain passing an array to a function.

An array can be passed to a function by keeping a parameter with an array tag with empty square brackets(like []). For example:

void function(int i[]) {..} /* parameter */

…

int k[10];

function(k); /* caller function */

What do you understand by a function prototype?

Declaration of the function is known as a function prototype and it consists of the function name, parameters list, and return type, but it does not consists of definition, that is, the function body; this is the same as declaring a variable but not defining it. For example:

int printf(const char *, int/);

How do you leverage a pointer to a function?

The toughest part about leveraging a pointer to a function is the declaration. For example, creating a pointer, pf, that points to the strcmp function where the strcmp functio n is declared as follows:

int strcmp(const char *, const char *)

To setup pf to point to the strcmp function, the declaration should look like the strcmp function's declaration, but should have *pf rather than strcmp :

int (*pf)(const char *, const char *);

Note the parentheses around *pf. Pointers to functions are interesting when you pass them to other functions.

A function that takes function pointers says, in effect, Part of what I do can be customized. Give me a pointer to a function, and I will call it when that part of the job needs to be completed.

When would you use a pointer to a function?

Pointers to functions are effective when you pass them to other functions. A function that takes function pointers, in effect means, part of what I do can be customized. Give me a pointer to a function, and I'll call it when that part of the job needs to be done. This is known as a callback. It leverages a lot in the graphical user interface, GUI libraries, in which the style of a display is built into the library but the contents of the display are part of the application.

Structures and unions

What will be the output of the following code snippet?

include

```
enum day {sunday = 1,monday,tuesday,wednesday,thursday = 3,friday,saturday};

int main()

{

  int i;

  printf("%d %d %d %d %d %d %d", sunday, monday, tuesday,

  wednesday, thursday, friday, saturday);

  scanf("%d",&i);

  return 0;

}
```

The answer is 1 2 3 4 3 4 5

Explanation: The enum assigns the value with the single increment. If a value is explicitly assigned to an element, it will just set that value to that element and start to increment from that assigned value for the following elements.

Pointers

What do you understand by a void pointer?

A void pointer is a special pointer, which can point to any other data type without the compiler knowing about it. This facilitates passing a pointer to a function of any type and this can be decided at runtime. For example, the input parameters x and y can be both an integer and a string:

```
Void voidPointer (int type, void *x, void *y, void *z)

{

  if(type == 1)/* int*/

  z = ((int) *x) + ((int)y);

  else /string/

  sprintf((char)z,"%s%s",(char)x,(char*y));

}
```

What is the value of NULL?

The value of NULL is 0 or (void*) 0. Whenever NULL has to be compared with a variable or assigned to a variable, depending on the type of that variable, the NULL value is decided.

char *p="SAMPLETEXT", *q ="SAMPLETEXT"; Are these two pointers equal ? If yes, then explain.

In C, strings(not array of characters) are immutable. This means that a string once created cannot be modified. Only flushing the buffer will remove it. Also, when a string is created, it is stored in the buffer. Next time, when a new string is created, it will check whether that string is present in the buffer. If it is present, that address is assigned, or else a new address stores the new string and this new address is assigned.

What is a dangling pointer in C?

If a pointer is pointing to a memory address of a variable, but the variable is deleted or does not exist in the current scope, then the pointer is called a dangling pointer. For example:

```
# include

int *function();

int main()

{

  int *ptr;

  ptr=func();

  printf("%d",*ptr);

  return 0;

}

int * function()

{

  int x=18;

  return &x;

}
```

The output is garbage value, since the variable x has been freed as soon as the func() function is returned. The following diagram depicts a

scenario where the objects are deleted from the memory and the pointer to that object has become a dangling pointer:

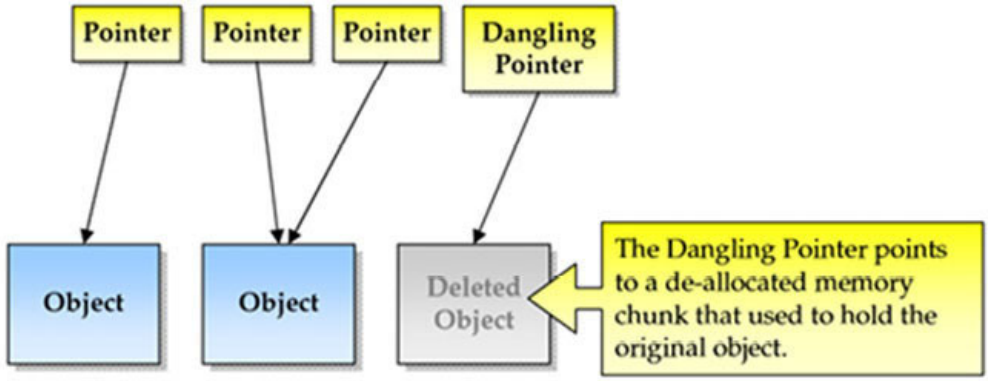


Figure 7.2: Dangling Pointer Scenario

What is a wild pointer in C?

A pointer that has not been initialized is known as a wild pointer in C. For example:

```
int main()
{
  int *ptr;
  printf("%u\n",ptr);
  printf("%d",*ptr);
  return 0;
}
```

The output isany address, garbage value.

Here, ptris a wild pointer because it has not been initialized. The wild pointer is not the same as a NULL pointer. Because the NULL pointer does not point to any location, but a wild pointer points to a specific memory location, and the memory may not be available for the current application, which is fatal.

What is the difference between a near pointer and far pointer?

Compilers leverage two types of pointers, which are as follows:

Near pointers are 16 bit long and can address a range of 64KB. Far pointers are 32 bit long and can address a range of 1MB. Near pointers operate within a 64KB segment. There's one segment for data and one segment for function addresses.

Far pointers have a 16-bit base - the segment address and a 16-bit offset. The base is multiplied by 16, so a far pointer is 20 bit long. For example, if a far pointer had a segment of 0x7000 and an offset of 0x1224, the pointer would refer to address 0x71224. A far pointer to a segment of 0x7122 and an offset of 0x0004 would refer to the same address.

Memory management

Can we declare the size of an array at runtime?

No, the size of an array must be declared at compilation time only. An alternative option is to utilize dynamic allocation APIs calloc or malloc.

What is the difference between calloc and malloc?

What do you understand by heap memory?

Malloc, calloc, and reallocget memory from the heap. Stack memory allocations process is faster than the heap allocation. Heap memory allocation is flexible than stack memory allocation. Memory can be allocated/deallocated in any order. This heap memory isn't deallocated automatically, the free() API is leveraged to deallocate the heap memory. The following diagram shows the application memory and the different sections of the memory, including stack and heap memories:

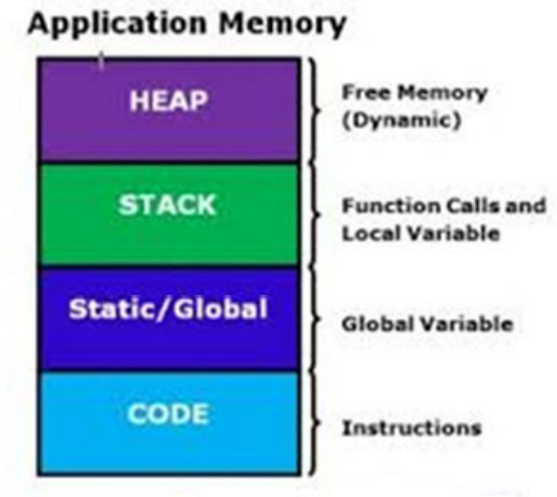


Figure 7.3: Application memory

What happens if you try to free a pointer twice?

It is hazardous to free the same memory twice and if the memory has not been re-allocated, it will generate a double free error.

How does free()method know about how much memory to release?

There's no predefined process and most systems, track the memory block as linked lists. When memory allocation functions are invoked, the memory blocks given to the caller are added to a linked list, and the size, block number, and relevant details are written on the header node. There is just a guideline and the process may differ from compiler to compiler environment.

How to print an address?

The best option is to leverage %p inprintf() or fprintf . The %p will instruct the compiler to utilize the best type, while printing the address according to the environment, as the size of a pointer varies from system to system.

How to assign one array to another array?

One cannot assign an array to another array. Arrays are not values, and they do not refer to just one variable, they are a set of variables. So, they cannot be placed on the left-hand side of an assignment operator. For example, the following code snippet will generate a compilation error:

Int x[5], y[5];

x = y;

What is the difference between a memory copy – memcpy and string copy - strcpy?

They both copy bytes from the source pointer to a destination pointer. However, the difference is that thestrcpyis specifically designed to copy strings, and it stops after getting the first \0 character. One needs to mention the length of the data to be copied, starting from the source pointer. memcpyis designed to work on all data types.

Miscellaneous

What is the order of operator precedence, right to left or left to right?

None of them is predefined or a standard and depends on the environment/compiler. C does not always start evaluating from right to left or left to right. Normally, function calls are done first, followed by complex operations and then simple operations. Hence, it is a best practice to leverage parenthesis for all the operations, without relying on precedence. The following table depicts the typical operation precedence with precedence level highest at the top and lowest at the bottom of the table:

Category	Operator	Associativity
Postfix	() [] -> . ++ - -	Left to right
Unary	+ - ! ~ ++ - - (type)* &sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Shift	<<>>	Left to right
Relational	<<= >>=	Left to right
Equality	== !=	Left to right
Bitwise AND	&	Left to right
Bitwise XOR	^	Left to right
Bitwise OR	|	Left to right
Logical AND	&&	Left to right

Figure 7.4: Operator precedence

What is the difference between X++ and ++X?

The ++ operator is known as an incremental operator. When the operator appears after the variable, then the expression is first evaluated, and then the variable is incremented by 1. When the operator appears before, the variable is incremented by 1 before it is leveraged in the expression.

What happens when we leverage the incremental operator in a pointer?

This depends on the type of pointer and gets incremented by the data type, the pointer is pointing to. For example:

char a; a++; /* a increments by 1*/

int b; b++; /* b increments by 4 - for 32 bit system */

What will be the output of the following code snippet?

int number1=5;

int number2=5;

number1 =number1++ + number1--;

number2 =++number2 + --number2;

printf("number1=%d number2=%d",number1,number2);

The output will be num1=10 num2=10 .

What does const keyword do?

Access specifier const, tells the compiler that the value of the variable is not going to be changed once it is initialized and the compiler enforces it throughout the lifetime of the variable.

When do we leverage typecast?

There are two main objectives of typecast:

Convert datatype X to a datatype Y . For example, if you typecast a float variable 1.45 to int , then it will be 1 .

Cast any pointer type to and from void * . This is utilized in generic functions such as memory copy, where the execution is an independent pointer

type.

What is the difference between declaring and defining a variable?

The declaration is made to tell the compiler the data type of the variable, and it inherently means that somewhere in the scope of the program, this variable is defined or will be defined.

Moreover, defining a variable means allocating memory for the variable on the stack. For example:

extern int declaration1; /* declaration */

int defination2; /* definition */

Is it possible to execute the code even after the program exits the main() function?

There is a standard C function atexit for this purpose to perform some operations when the program is exiting. One can register functions with atexit to be executed at the time of termination. For example:

include

include

void *some*FUNC_(void);

int main(int argument-c, char** argument-v)

{

  …

  atexit(*some*FUNC_);

  ….

}

Is leveraging exit() similar to using return?

No, they are not the same. The return statement returns the control to the caller function, that is, it exits from the lowest level of the call stack. Whereas, the exit statement makes the program return to the system from where the application had started. Exit always exits from the highest level in the call stack. In case there is only one level of function, calling then both will do the same.

What is the difference between a global variable and static variable?

The global variable can be accessed by any function, including the ones from different files. A static variable declared is accessible to all the functions defined in the same file or translation unit. The following diagram depicts the different types of variables that can be defined and declared in C programs:

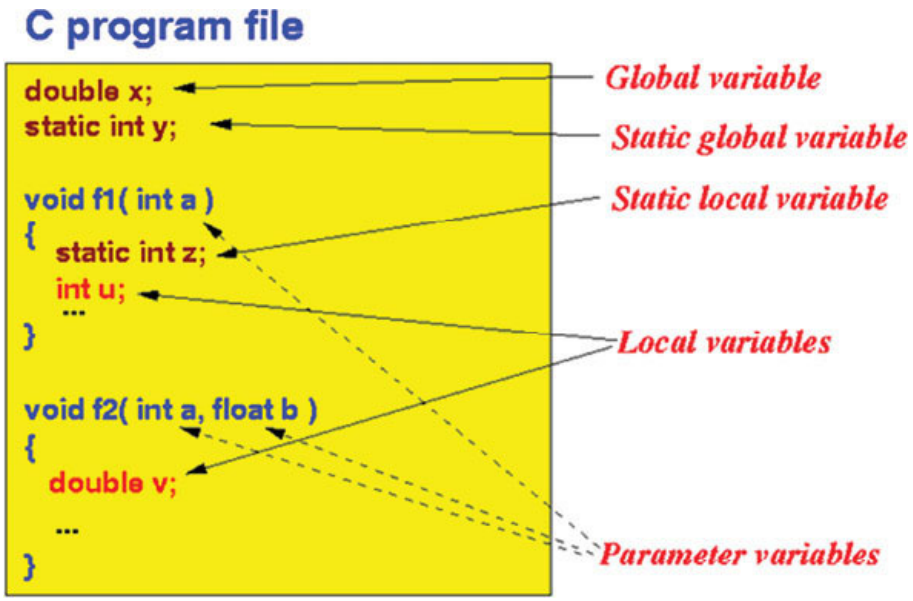


Figure 7.5: Difference types of variables

Write a C program to swap two variables without leveraging a third variable.

include

```
int main()
{
  int x=5,y=10;
  x=y+x;
  y=x-y;
  x=x-y;
  printf("x= %d y= %d",x,y);
}
```

Write a C program to find the size of a structure without using the sizeof operator.

```
Struct XXX
{
  int x;
  float y;
  char z;
};
int main()
{
  struct XXX *ptr=(struct XXX *)0;
  ptr++;
  printf("Size of structure is: %d",*ptr);
```

```
  return 0;
}
```

What is the output of the following code snippet?

```
# include
int main()
{
  printf("%d", printf("%d", 1234));
  return 0;
}
```

The answer is 12344. 1234, which will be printed by the second printf and it will return 4 as printf returns the number of letters it printed.

How can you determine the size of an allocated portion of memory?

You can't, really. The free() method can, but there's no way for your program to know the trick free() uses. Even if you disassemble the library and discover the trick, there's no guarantee the trick won't change with the next release of the compiler.

Is exit() the same as return()?

The exit() is leveraged to exit the program and return the control to the operating system. In contrast, the return statement is leveraged to return from a function to the calling function. If you invoke a return from the main function, you are essentially returning control to the calling function, which is the operating system. In this case, the return statement and exit function in the same manner.

C++ programming

Core concepts

What is the difference between C and C++?

C++ is not a pure OOP language, but supports both procedural and object-oriented while C is a procedural programming language.

C follows top-down approach, that is, a solution is created in step-by-step manner, whereas C++ follows a bottom-up approach, that is, where base elements are established first and are integrated to create complex application.

The C language does not provide adequate security mechanism, but in C++, one can leverage access specifiers for the class members to make it inaccessible to the outside world.

C++ allows leveraging functions in structures, but C does not permit.

C++ provides function overloading while C does not provide function overloading.

C does not have a built-in exception handling capability, though one can emulate it whileC++ supports exception handling.

C++ supports reference variables, that is, two variables can point to the same memory location whereas C does not support it.

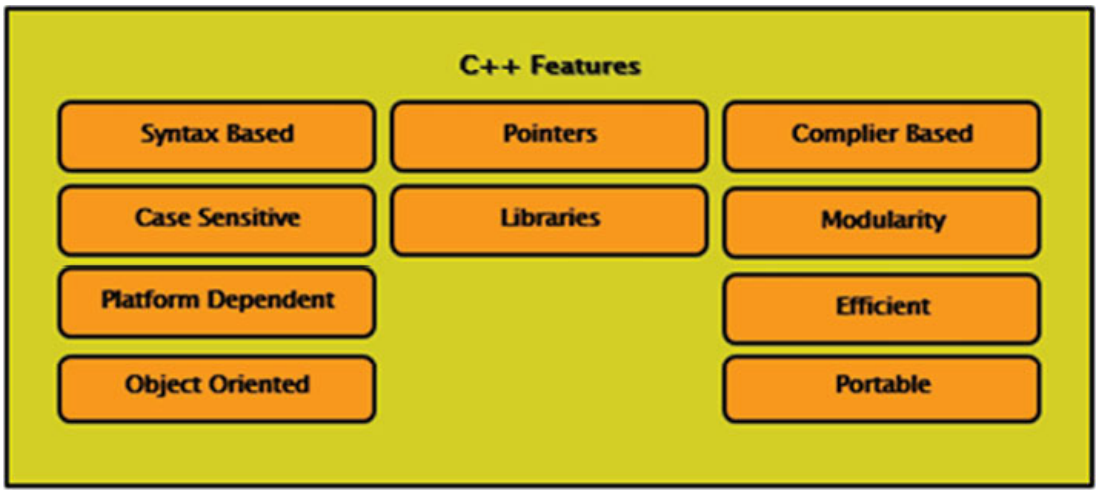


Figure 7.6: C++ language features

What is a class?

The class defines a datatype; it's a type definition of the category of thing(s). But a class actually does not define the data; it just specifies the structure of data. To leverage them, one needs to create objects out of the class. The class is a blueprint of a building, and one needs to construct the building(s) out of that plan. One can create any number of buildings from the blueprint. Similarly, one can create any number of objects from a class. The following code snippet describes the process to declare and define a class:

```
class Vehicle

{

  public:

  int numberOfTyres;

  double engineCapacity;

  void drive(){

    // code to drive the car

  }

};
```

What is an object/instance?

The object is the instance of a class, which is a concrete entity. From the preceding example, we can create an instance of class Vehicle as follows:

```
Vehicle vehicleObject;
```

One can have different objects of the class Vehicle ; for example, we can have Vehicle objects with 2 tyres, 4tyres, and more. Similarly, various engine capacities.

What do you understand by access specifiers?

Access specifiers are leveraged to define the members, that is, functions and variables will be accessed outside that class. There are three types of access specifiers:

public: Members declared as the public are accessible from anywhere in the program.

private: Private members are accessible only within the same class, and they cannot be accessed outside that class.

protected: Protected members cannot be accessed outside the class except a derived/child class.

The following diagram depicts the access specifier table for public, protected, and private access modifiers:

Figure 7.7: Access specifiers

What are the basic concepts of OOP?

Classes and objects: Refer to the earlier questions for the concepts about classes and objects.

Encapsulation: Encapsulation is the mechanism by which data and the associated methods are bound together and the data hidden from the outside world. It's also known as data hiding. In C++, encapsulation is achieved leveraging the access specifiers such as public, private and protected. Data members will be declared as private, thus protecting them from direct access from outside and public methods will be provided to access this data. For example:

```
class Person
{
  private:
    int age;
  public:
    int getAge(){
      return age;
    }
    int setAge(int value){
      if(value > 0){
        age = value;
      }
    }
};
```

In the Person class, access to the data member age is protected by declaring it asprivateand providing public access methods. What would happen if there were no access methods and the field age was public ? Anybody who has a Person object can set an invalid value; for example, negative or very large value for the age field. So, by encapsulation, we are preventing direct access from outside, and thus have complete protection, control, and integrity over the data.

Data abstraction: Data abstraction refers to hiding the internal implementations and publishing only the required details to the outside world. In

C++, data abstraction is implemented using the abstract classes and interfaces. For example:

```
class Stack
{
  public:
    virtual void push(int)=0;
    virtual int pop()=0;
};
class MyStack : public Stack
{
  private:
    int arrayToHoldData[]; //Holds the data from stack
  public:
    void push(int) {
     // implement push operation using array
    }
    int pop(){
     // implement pop operation using array
    }
  };
```

In the preceding example, the outside world only needs to know about the Stack class and the push and pop operations. Internally, a stack can be implemented leveraging arrays or linked lists or queues. This means, as long as the push and pop method performs the operations as expected, you have the freedom to change the internal implementation without affecting other applications that leverage the Stack class.

Inheritance: Inheritance facilitates one class to inherit properties of another class. In other words, inheritance allows one class to be defined in terms of another class. For example:

```
class SymmetricShape
{
  public:
    int getSize()
    {
     return size;
    }
    void setSize(int w)
    {
     size = w;
    }
    protected:
```

```
    int size;
  };
// Derived class
class Square: public SymmetricShape
{
 public:
   int getArea()
   {
    return (size * size);
   }
 };
```

In the preceding example, the Square class inherits the properties and methods of the SymmetricShape class. Inheritance is one of the key concepts in C++/OOP. It helps to modularize the code, improve reusability, and reduces tight coupling between components of the application.

In how many ways, can we initialize an int variable?

In C++, variables can be initialized in two ways: the traditional C++ initialization leveraging = operator and leveraging the constructor notation.

Traditional C++ initialization: int i = 10

Leveraging C++ constructor notation: int i(10)

What are inline functions?

The inline functions are special functions in C++ for which the compiler replaces the function call with the function body. Inline functions make the program execute faster than the normal functions since the overhead of managing the current state to the stack is eliminated. By providing flexibility of making a function inline, one can optimize the program code. This is just a directive to the complier and in practice, it is the compiler's decision whether to make a function inline or not, regardless of the directive. The following code snippet describes the way to define and leverage an inline function:

```
inline int min(int a, int b)
{
 return (a < b)? a : b;
}
int main( )
{
 cout << "min (20,10): " << min(20,10) << endl;
 cout << "min (0,200): " << min(0,200) << endl;
 cout << "min (100,1010): " << min(100,1010) << endl;
 return 0;
}
```

If the complier decides to make the minfunction inline, then the preceding code snippet will be as follows:

```
int main( )
```

```
{
  cout << "min (20,10): " << ((20 < 10)? 20 : 10) << endl;
  cout << "min (0,200): " << ((0 < 200)? 0 : 200) << endl;
  cout << "min (100,1010): " << ((100 < 1010)? 100 : 1010) << endl;
  return 0;
}
```

What is a translation unit in C++?

The C++ programs are organized into different source files (.cpp, .cxx , and so on). When one considers a source file, at the preprocessing stage, extra content may get added to the source code such as the contents of header files and some content may get removed such as the part of the code snippet in the #ifdef or #ifndef block will resolve to false /0 based on the symbols defined. This consolidated content is called a translation unit.

What is internal and external linking?

A symbol is linked internally when it can be accessed only from within the scope of a translation unit. In external linking, the symbol can be accessed from other translation units as well. This linkage is controlled utilizing static and extern keywords. The following diagram depicts the stages the programs goes through which starts with the preprocessor, then the complier, then the assembler and lastly the linker before an executable is created out of the source code and libraries:

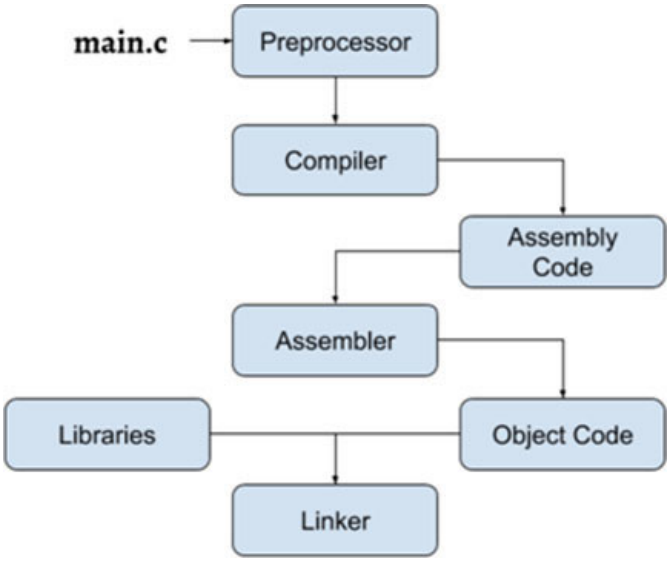


Figure 7.8: Different stages of a program

Keywords and operators

What is the use of the volatile keyword?

Most compilers perform optimization to the program code to speed up the application. For example:

```
int a = 10;
while( a == 10){
    // Do something
}
```

The compiler may decide that the value of a is not getting changed from the program and replaces it with while (true) which will result in an infinite loop. In a typical business application, the value of a will be updated from outside the program/translation unit. The volatile keyword is a directive to the compiler that the variable declared using volatile might be leveraged from outside the current scope, so the compiler should not perform optimization. This typically is the case with multi-threaded programs.

Constructors and destructors

What is a copy constructor?

The copy constructor is a special constructor, which is leveraged to create a copy of an object. The compiler will provide a default copy constructor if one is not defined in the implementation. This implicit constructor will copy all members of the source object to a target object.

Implicit copy constructors are not recommended because if the source object contains pointers, they will be copied to the target object, causing heap corruption when both the objects with pointers refer to the same location. In this case, it is a best practice to define the custom copy constructor and perform a deep copy. The following code snippet describes a way to define a copy constructor and when it can be invoked in the program:

```
class SampleClass{

  public:

    int* ptr;

    SampleClass();

    // Copy constructor declaration

    SampleClass(SampleClass &obj);

};

SampleClass::SampleClass(){

  ptr = new int();

  *ptr = 5;

}

// Copy constructor definition

SampleClass::SampleClass(SampleClass &obj){

  //create a new object for the pointer

  ptr = new int();

  // Now manually assign the value

  *ptr = *(obj.ptr);

  cout<<”Copy constructor…\n”;

}
```

What are virtual functions?

Virtual functions are class members, which are declared by leveraging the virtual keyword. When base class type references are initialized by leveraging a derivedclass object, an overridden method declared as virtual is invoked using the base class reference; the method in the derived class object is invoked. The following code snippet describes a way to declare, define and leverage virtual functions in C++:

```
class Base

{

    int a;

    public:

      Base()

      {
```

```
        a = 1;
      }
      virtual void method()
      {
        cout << a;
      }
};
class Child: public Base
{
   int b;
   public:
      Child()
      {
        b = 2;
      }
      virtual void method()
      {
        cout << b;
      }
};
int main()
{
   Base *pBase;
   Child oChild;
   pBase = &oChild;
   pBase->method();
   return 0;
}
```

In the preceding example, even though the method is invoked on the base class reference, the method of the derived class will get invoked since it's declared as virtual.

What do you mean by pure virtual functions in C++?

The pure virtual function is a function that doesn't have an implementation and needs to be defined by the immediate non-abstract derived class. A class becomes an abstract class if there is at least one pure virtual function, and thus pure virtual functions are leveraged to create interfaces in C++. A function is made a pure virtual function by the using a signature = 0 , which is appended to the function declaration as shown in the following code snippet:

```
class SymmetricShape {
```

```
 public:
   // draw() is a pure virtual function.
 virtual void draw() = 0;
};
```

Why are pure virtual functions used if they don't have an implementation?

Pure virtual functions are leveraged when it doesn't make sense to provide a definition of a virtual function in the base class, or an appropriate definition does not exist in the context of the base class. Consider the preceding example. The SymmetricShape class is used as the base class for shapes with symmetric structure, (square, circle, and equilateral triangle). In this case, there exists no proper definition for the draw() function in the SymmetricShape base class;instead, the derived classes of SymmetricShape (circle, square, and more) may implement this method and draw the proper shape.

What are virtual destructors?

Virtual destructors are leveraged for the similar purpose as virtual constructors. When you remove an object of the baseclass, which is referenced by a parent class pointer, the destructor of only base class will be invoked. But if the destructor is defined using the virtual keyword, both the destructors, that is, of the parent and childclass will get invoked.

Inheritance and polymorphism

What do you mean by early binding and late binding?

Binding is the process of linking the actual address of functions to their references. There are two mechanisms, which are as follows:

During compilation, that is, early binding: For all the direct function references, the compiler will replace the reference with the actual address of the method.

At runtime, that is, late binding or dynamic binding: In the case of virtual function calls, leveraging a base class reference; for example, the compiler does not know which method will get invoked at runtime. In this case, the compiler will replace the reference with a code snippet to get the address of the function at runtime.

Miscellaneous

What are storage classes? How many storage classes are available in C++?

The storage class is leveraged to specify the visibility, scope and lifetime of symbols both variables and functions. This means that the storage classes specify where the function or the variable can be accessed and till what timeframe these variables will be available during the execution of the application. The following lists describe the different storage classes available in C++:

auto: This is the default storage class for variables. It can be accessed only from within the declaration scope. The auto variables are allocated at the start of the enclosing block and deallocated at the end of the enclosing block.

register: These are similar to auto variables. The difference is that register variables might be stored in the CPU/processor register instead of RAM, which means the maximum size of the register variable should be the size of the CPU register (for example, 16bit, 32bit, or 64bit). This is normally leveraged for frequently accessed variables like counters to improve performance. But note that declaring a variable as a register does not mean that they will be stored in the register as it's just a directive.

static: A static variable will be kept active until the end of the program unlike creating and destroying every time they move in and out of the scope. This helps to maintain their values even if the control goes out of the scope. When the static variable is leveraged with global variables, they will have internal linkage, that is, and can’t be accessed by other source files. When the static variable is used in case of a class member, it will be shared by all the objects of a class instead of creating separate copies for each object.

extern: A extern is leveraged to tell the compiler that the symbol is defined in another translation unit such as source files and not in the current one, that is, the symbol is linked externally. The extern symbols have static storage duration, which is accessible throughout the program life. Since no storage is allocated for the extern variable as part of the declaration, they cannot be initialized while declaring.

mutable: The mutable storage class can be leveraged only on non-static, non-const data member of a class. The mutable data member of a class can be modified even if it is part of an object, which is declared as const.

The following table is the summary of C++ storage class specifiers:

What is realloc() and free()?

void* realloc (void* ptr, size_t size): This function is leveraged to change the size of the memory object pointed by the ptraddress to the size given bysize. If ptr is a null pointer, then realloc will behave like malloc() . If the ptr is an invalid pointer, then the defined behavior may occur depending on the implementation. Undefined behavior may occur if the ptr has previously been deallocated by free() , or dealloc() or ptr do not match a pointer returned by a calloc(), malloc(), or realloc() .

void free (void* ptr): This function is used to deallocate a block of memory that was allocated using malloc(), calloc(), or realloc() . If ptr is null, this function does not do anything.

Explain the difference between shallow copy and deep copy.

In a shallow copy, all the fields of the source object are copied to the target object. This means that if there is a dynamically created field in the source object, a shallow copy will copy the same pointer to the target object. So, you will have two objects with fields that are pointing to the same memory location which is not a best practice.

In a deep copy, instead of copying the pointer, the object itself is copied to the target. In this case, if ones modify the target object, this will not affect the source object. By default, the copy constructors and assignment operators do a shallow copy. To ensure a deep copy, one needs to provide a custom copy constructor and override the assignment operator.

The following diagram depicts the shallow copy vs deep copy process:

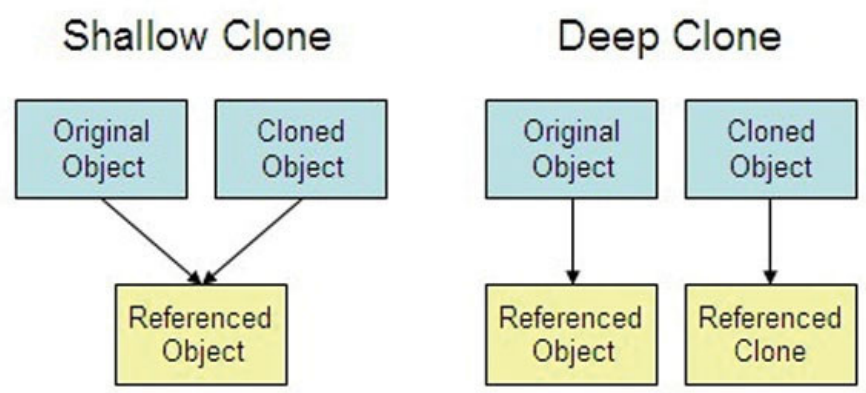


Figure 7.9: Shallow copy vs deep copy

What do you mean by persistent and non-persistent objects?

Persistent objects can be serialized and written to a disk or any other stream. So, before closing the application, one would be able to serialize the object, and on a restart, one can deserialize it. For example, drawing applications usually leverage serializations. Objects that cannot be serialized are called non-persistent objects.

Is it feasible to get the source code from a binary file of an application?

Technically, it is feasible to generate the source code from a binary file of the application and this process is known as reverse engineering. There are various reverse engineering tools available currently. But in practice, most of these tools may not re-generate the entire exact source code because the information is lost due to compiler optimization.

What is meant by a reference variable in C++?

In C++, a reference variable allows you to create an alias, that is, the second name for an already existing variable. A reference variable can be leveraged to access read/write the original data. This means that both the variables and reference points to the same memory location. In effect, if one changes the value of a variable using the reference variable, both will get changed as both refer to the same memory location. Appending an ampersand (&) to the end of datatype makes a variable eligible to be used as a reference variable. The following code snippet describes a way to leverage reference variables:

```
int a = 20;

int& b = a;
```

The first code snippet initializes an integer variable a . The second code snippet creates aninteger reference initialized to variable a . For example,

to see the working of reference variables:

```
int main ()
{
  int a;
  int& b = a;
  a = 10;
  cout << “Value of a : “ << a << endl;
  cout << “Value of a reference (b) : “ << b << endl;
  b = 20;
  cout << “Value of a : “ << a << endl;
  cout << “Value of a reference (b) : “ << b << endl;
  return 0;
}
```

The preceding code snippet will create the following output:

Value of a: 10

Value of a reference (b): 10

Value of a: 20

Value of a reference (b): 20

What is the difference between reference variables and pointers in C++?

What is a design pattern?

A design pattern is a general reusable solution to a commonly occurring problem within a given context in software design. A design pattern is not a finished design that can be transformed directly into source or machine code. It is a description or template on how to solve a problem that can be used in many different situations. Patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system.

What all are the typesof design patterns?

There are three types of design patterns:

Creational patterns: This type of pattern addresses problems of creating an object and separating it from operations.

Structural patterns: This type of pattern addresses problems of using object oriented constructs to organize classes and objects.

Behavioral patterns: This type of pattern addresses problems of assigning responsibilities to classes.

What is Unified Modeling Language (UML)?

Unified Modeling Language (UML) is a standard language for designing and documenting a system in an object-oriented manner. It has nine diagrams, which can be used in designdocument to express a design of software architecture.

Conclusion

This section provides a comprehensive C/C++ interview Q&A. Irrespective of the domain or language, one pursues, More than 90% of campus interviews questions will be from C or C++ or Java programming languages. The motivation is that the candidates fresh out of colleges are not expected to be experts of widely leveraged languages like Java, C++, Python, and more. Most universities and colleges will have C or C++ or Java as a part of the curriculum and are aligned to the IT industry demands. For example, C language is being adopted along with data structure and C++ as part of object-oriented programming stream in most engineering colleges.

This section covered the C and C++ programming languages question bank. The section on C had questions bank for aspects pertaining to preprocessing, functions, structures, pointer and memory management. The section on C++ had a question bank for aspects pertaining to core concepts, keywords and operators, constructors and destructors, inheritance and polymorphism. The next chapter presents an exhaustive question bank for Java programming language with special emphasis on practical scenarios and business cases. The chapter will cover aspects pertaining to Java iterator, classes, exceptions, serialization, and polymorphism.

CHAPTER 8

Java Programming

Aprogramming language consists of set of instructions that can be customized to yield various business outcomes. Programming languages consist of set of instructions for a machine and are utilized to create applications that solve specific domain problems.

A number of programming languages already exist and many more are developedeach year. A programming language describes two basic key aspects: one is syntax (form) and second is semantics (meaning). A few languages are defined by specification artifacts; for example, C programming language is specified by ISO while other languages such as Perl have a dominant implementation that is treated as a reference. Some languages have both, the basic language reference and extensions from dominant implementation.

This chapter presents an exhaustive question bank on Java programming language with special emphasis on practical scenarios and business cases. The chapter provides various sections that cover aspects related to core concepts, keywords and operators, classes, interfaces, construction, destructors, inheritance, and polymorphism. The chapter also covers miscellaneous key aspects, including Java iterator, classes, exceptions, JVM, IO, collections, and serialization

Core concepts

What is Java? What are the objectives of Java?

Java is an object-oriented programming language that provides application development and deployment environment. Java programming language is very similar to C++ language. The Java development environment provides tools and frameworks for interpreting, compiling, packaging and deploying Java applications. The following diagram depicts the key characteristics of Java programming language:

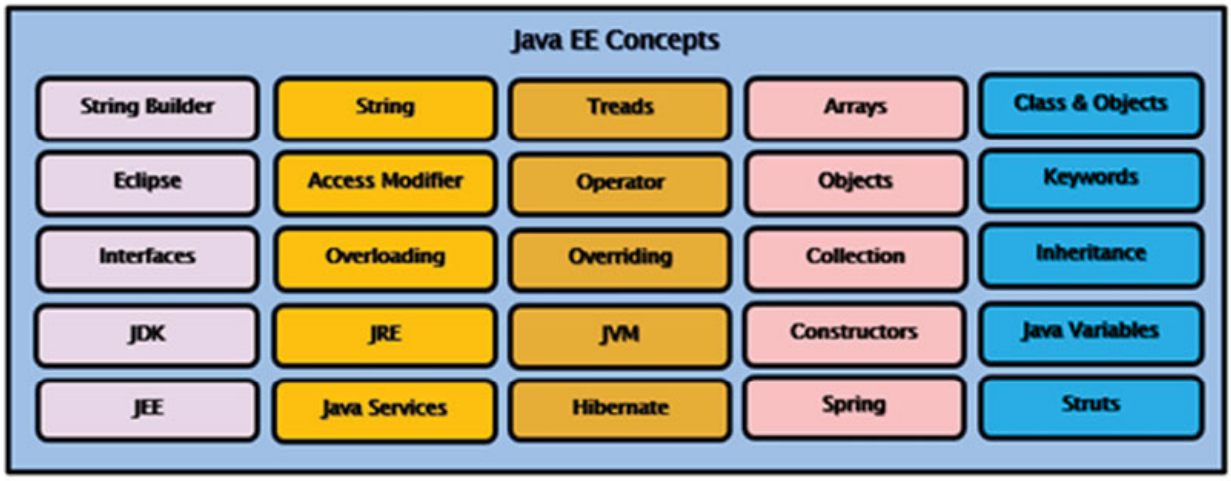


Figure 8.1: Java EE concepts

The objectives of Java technologies are as follows:

Simple: The syntax is based on C++ language. Even though the C++ syntax was adopted, some features that were cryptic were removed to make Java simpler.

Object-oriented: The object-oriented aspects of Java are similar to C++ programming. A major difference between Java and C++ is multiple inheritances. While C++ supports multiple inheritance, Java does not directly.

Architectural neutral: A Java compiler generates an intermediate bytecode, which doesn't depend on the operating system. Therefore, Java programs can run on any hardware, irrespective of the OS and hence, it's architectural neutral.

Portable: The size of datatypes is always the same, irrespective of the underline system architecture. An int in Java will always be 32 bit unlike in C or C++, where the size may be 16- or 32-bit depending on the compiler or machine. The Java framework enables porting its application to any

systems like UNIX, Windows, or Macintosh.

Distributed: Java's networking capabilities are both strong and flexible. Java applications are capable of accessing objects across the network and local file system.

Secure: Since Java is mostly leveraged in a networked environment, emphasis has been put on security to enable creation of virus and tamper-free applications.

Multithreaded: This is the ability to perform more than one task at a time in an application. Compared to other languages, it is easier to implement multi-threaded applications leveraging Java frameworks and libraries.

Performance: The bytecodes can be translated at runtime into machine code for the specific CPU on which the application it is running, thus providing excellent performance.

What are the key components of the Java framework?

The key components of the Java framework are as follows:

Java Runtime Environment (JRE)

Java Virtual Machine (JVM)

Just-in Time Compiler

Java tools

Garbage collector

What is a Java Virtual Machine (JVM)? What are the advantages of a JVM?

A Java Virtual Machine (JVM) is a self-contained operating environment that behaves as a separate computer. The JVM along with Java Class Libraries that implement the Java application programming interface (API) form the Java Runtime Environment (JRE) .

The JVM has two advantages:

System independence: The Java application will run in a consistent manner on any Java VM, regardless of the hardware and software.

Security: Because the JVM has no contact with the operating system, there is little possibility of a Java program intruding into other applications.

What are CLASSPATH variables?

The CLASSPATH variable is an environment variable that tells the JVM where to look for user-defined classes and packages of an application. When a Java program is being executed, the JVM looks for the required classes in the locations specified by the CLASSPATH variable. Classes are loaded into the JVM, only when a call is made to leverage the content of the class objects, also known as Lazy Loading .

Explain the architecture of the code execution process inside the JRE.

Java compilers compile the Java source code into the Java bytecode. The JVM interpretsthe Java bytecode, sends the necessary commands to the underlying hardware, or the just-in-time (JIT) compiler to the native machine code and executes it directly on the underlying hardware. Most JVM's leverage JIT compilation, which provides the execution speed equivalent to C/C++ applications. Most widely used JVM is Oracle's Hotspot, which is written in C++ language. The JVM along with Java Class Libraries that implement the Java API forms the JRE. The following diagram depicts the code execution process in the JRE:

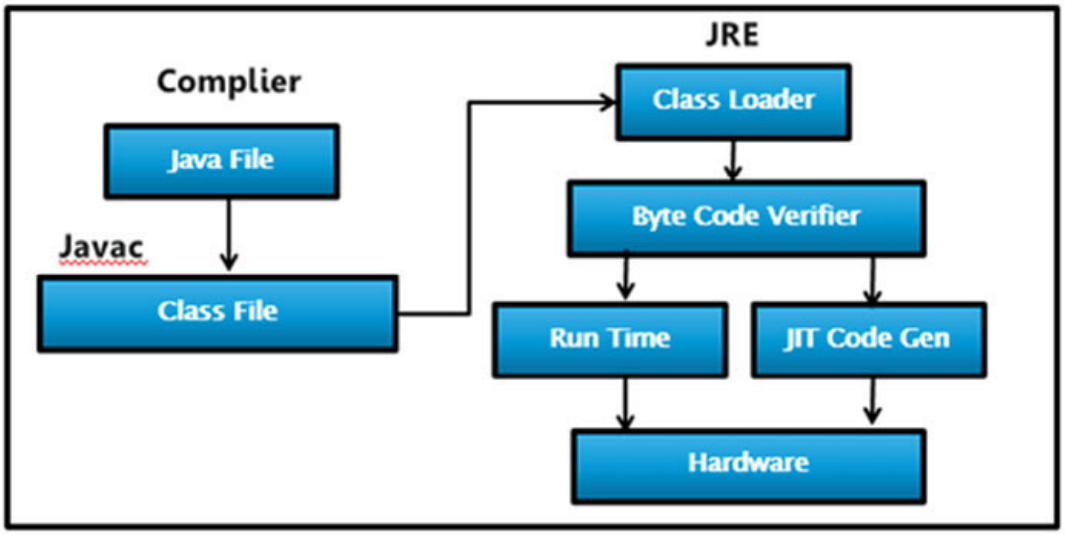


Figure 8.2: Code execution process in JRE

What are the steps involved in the Java application execution?

The following are the steps in the Java application execution process:

Compiling Java source files into a * .class file - bytecode

Loading the class file into the JRE using the class loader

Use the Bytecode verifier to check for bytecode specifications

Use the Just-in Time code generator or interpreter for bytecode execution

Explain JIT, JDK, Garbage collector, Class Loader, Java API.

JIT compiler: The JIT compiles the Java bytecode to the native machine code and executes it directly on the underlying hardware. The JIT compilation provides the application execution speed that is equivalent to C or C++ applications.

JDK: The JDK is a superset of the JRE, which contains the JRE, compilers, and debuggers required to develop and deploy Java applications.

Garbage collector (GC): This application allocates memory at runtime for executing programs. After execution of programs, unused memory needs to be de-allocated. Programmers who develop applications for C/C++ programming language do allocation and de-allocation of memory. Java internally uses a garbage collector to deallocate unused memory, which makes the life of the programmers easier.

Class loader: The class loader is leveraged to load all the classes required to execute the application into the JVM. Once the classes are loaded, the memory required for the application is determined.

Java API: An API is a library of functions that Java provides for programmers/designers for common tasks like data structures, networking, file transfer, and so on. The examples of Java APIs are as follows:

Java.applet: Applet class

Java.awt: Windows, buttons, mouse, and so on.

Java.awt.image: Image processing

Java.io: System.out.print

Java.net: Sockets

Java.lang: Length method for arrays; exceptions

Java.util: System.getProperty

How to locate the starting method of a program?

The method starting with the following code snippet is the starting method of the Java application:

```
public static void main (String args[])

{

}
```

What do you understand by information hiding in Java?

In order to avoid direct access to attributes/parameters of a class, attributes/parameters are defined as private and the getter and setter methods are leveraged to restrict the accessto these attributes/parameters. In the following example, the FirstName class variable is the information hiding aspect where the data is not directly exposed to the outside world and it is managed via the setFirstName() and getFirstName() class methods:

```
public class Person {

  private String FirstName;

  public String getFirstName()

  {

    return FirstName;

  }

  public void setFirstName(String FirstName)

  {

    this.FirstName = FirstName;

  }

}
```

What do you understand by an interface?

An interface is a conceptual entity. It can contain only final variables and abstract methods. If we want to enforce a certain behavior on objects, so that the object provides certain functionalities irrespective of its underlying implementation, we use an interface.

What is an abstract class? What are the properties of an abstract class?

An abstract class is a conceptual class. An abstract class cannot be instantiated i.e. objects cannot be created from an abstract class. Abstract classes act as a base for a group of classes, which are closely related to a package. If a class is declared as final, we can't extend that class. In some cases, properties of a class should be extended and used in subclasses. Such classes are called abstract classes . A class that has one or more abstract methods is automatically abstract, and it cannot be instantiated.

The properties of abstract classes are as follows:

A class declared abstract with no abstract methods cannot be instantiated.

A subclass of an abstract class can be instantiated if it overrides all abstract methods.

A subclass that does not implement all the superclass abstract methods is itself an abstract class.

One cannot declare an abstract constructor or abstract static methods.

What is an object class in Java? What are the methods of an object class?

Every class created in Java extends the class Object by default. This means the Object class is at the top of the hierarchy. Due to this default inheritance of the object class, we can pass an object of any class as an argument to any method. The methods in the Object class are as follows:

equals( ): This method returns if both the objects are equal.

getClass( ): This method returns the class to which the object belongs.

finalize( ): This method is called when an object's memory is destroyed.

notify( ): This method isused to give messages to synchronized methods.

hashCode( ): This method returns the hashcode of the class.

notifyAll( ): This method is used to notify all synchronized methods.

wait(): This method suspends a thread.

toString(): This method returns the string equivalent of the object name.

wait(…): This method suspends a thread for a specified time in seconds.

What are the different scopes for Java variables?

The scope of a Java variable is determined by the context in which the variable is declared. Thus, a Java variable can have one of the three scopes:

Instance: These are typical object level variables. They are initialized to default values at the time of creation of an object and remain accessible as long as the object is accessible.

Local: These variables are defined within a method. They remain accessible only during method execution. When the method finishes the execution, these variables go out of scope.

Static: These are the class level variables. They are initialized when the class is loaded by the JVM for the first time and remain there as long as the class remains loaded. They are not tied to any particular instance of object.

What is the default value of the local variables?

The local variables are not initialized to any default value, neither primitives nor object references. If you try to use these variables without initializing them explicitly, the Java compiler will not compile the code. It will complain about the local variable not being initialized.

What is serialization?

Serialization is a methodology by which one can save the state of an object by converting it to a byte stream. The class whose instances need to be serialized must implement the Serializable interface. Then, you pass the instance to the ObjectOutputStream , which is connected to afile output stream that will save the object to a file. One should make sure that all the included objects are also serializable. If any one of the objects is not serializable, then it throws a NotSerializableException. The following diagram depicts the serialization and de-serialization process:

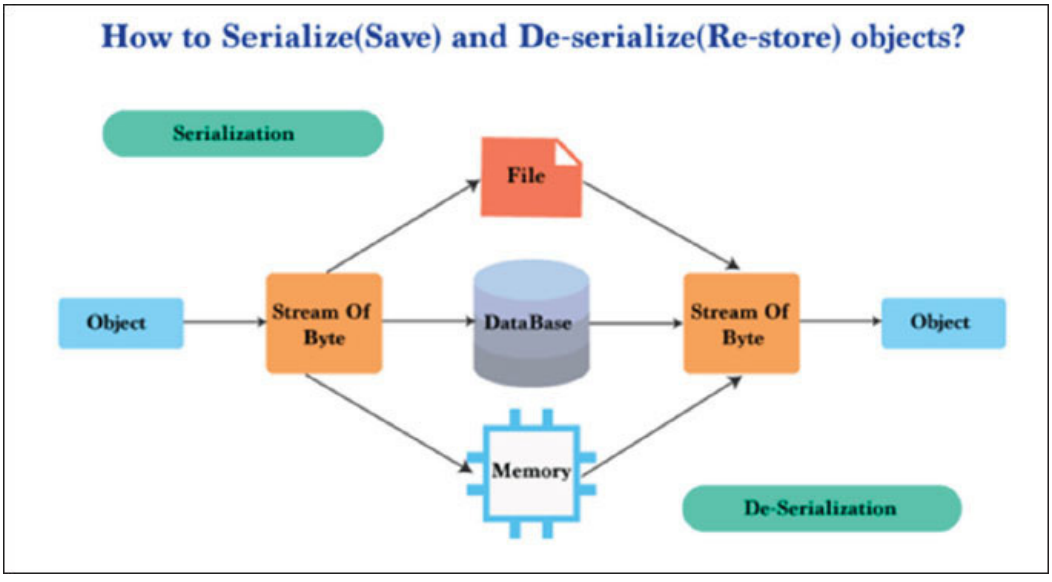


Figure 8.3: Serialization vs de-serialization process

What are wrapper classes?

Java provides specialized classes for each of the primitive data types. These are called wrapper classes. For example, a character, integer, double, and so on. It is easier to deal with primitives as objects. Besides, most of the collection classes store objects and not primitive types. Also,

the wrapper classes provide many utility methods due to which we need wrapper classes. Since we create instances of these classes, we can store them in the collection classes and pass them as collections. We can pass them around as method parameters. The following table depicts the different wrapper classes that are provided in Java language:

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

Figure 8.4: Wrapper classes

Can an application have multiple classes having the main() method?

Yes, one can have the main() methods in multiple classes. While starting the application, one has to mention the class name to be executed. The JVM will look for the main() method only in the class whose name is mentioned in the CLI. Hence, there is no conflict among the multiple classes having the main() method.

What is garbage collection in Java?

The objective of garbage collection is to identify and discard objects that are no longer needed by the application so that the resources can be reclaimed and re-used. A Java object is subject to garbage collection when it becomes unreachable to the program. Garbage collection does not guarantee that the program will not run out of memory. It is possible for programs to utilize the memory resources faster than they can be garbage collected. It is also possible for programs to create objects that are not subject to garbage collection.

What are Generic classes and what are the benefits of Generics?

Generic classes are one of the best features of Java 5. Since it is backward compatible with Java 1.4, old programs will work on the latest Java. Letus look at the features and use of generic classes. It's quite common that Java throws the CastException class at runtime while working with collection classes. To avoid this nasty surprise, Java has introduced compile-time type checking. If we are using generic classes, we don't need to typecast while retrieving the items from the collections. It’s advantage is that collections are type safe.

For example, you can create a Stack that holds only Strings as follows:

Stack names = new Stack();

Youcan create methods that need a type-safe collection as follows:

void printNames(Stack names) {

String nextName = names.pop(); // no casting needed!

names.push("Hello"); // works just the same as before

names.push(Color.RED); // compile-time error!

Name eight primitive Java types.

The eight primitive types are a byte, char, short, int, long, float, double, and Boolean.

Why is the main() method declared static? What is the argument of the main() method?

The main() method is invoked by the JVM/runtime even before the instantiation of class and hence, it is declared as static.

The main() method accepts an array of String as an argument. Also, void should always be before main() but the order of public and static declaration doesn't matter.

Does the order of public and static declaration matter in the main() method?

The order of public and static declaration doesn't matter, but void should be before main() .

What is the difference between Path and Classpath?

Path and Classpath are both operating system environment variables. Path is leveraged to specify the location of executables .exe files and Classpath is leveraged to specify the location .class files.

How to define a constant variable?

A variable in the Java program needs to be declared as static and final. This ensures only one copy of the variable exists and the value can't be changed. For example:

static final int MAX_LENGTH = 50; // Example for constant.

What are final, finally, and finalize?

final() is a access specifier which is applied to a class, method or variable. The finalvariable can't be changed, the finalmethod can't be overridden, and the finalclass can't be inherited.

finally{} is an exception handling block in the application which is invoked whether an exception is raised or not from the try block.

finalize() is a method of the Object class which will be invoked by the JVM before garbage collecting the object to give it a final chance for releasing resources.

When is a method defined as static? What is the importance of a static variable?

When a method needs to be accessed before the creation of an object of the class, it should be declared asstatic. A static method should not refer to instance variables without creating an instance of a class and it cannot leverage this operator to refer to the class instance.

Static variables are class level variables and all the objects of the class refer to the same variable. If one object changes the value, then the change gets reflected in all the objects. Static variables cannot be declared inside a method, and if it's declared, the class will not compile.

How can you print "Hello" even before main() is executed ?

Add a print statement inside a static code block. Static blocks are invoked by the JVM/runtime when the class is being loaded in memory and before the creation of an object. Hence, it will be executed before the main() method and will be executed only once.

Keywords and operators

What is "this" reference in Java?

To refer to the current object, this keyword is used. When we want to pass the instance of the current object to a method in another object, we pass the current instance as follows:

otherobj1.anymethod(this);

If a variable of an object and the parameter name of a method in the object are the same, we use this keyword to avoid ambiguity as follows:

```
String name;

public void method1(String name){

  this.name=name;

}
```

The this keyword included in parenthesis, that is, this() with or without parameters is used to call another constructor. The default constructor for the Car class can be redefined as follows:

```
public Car(){

  this(“NoBrand”,””NoColor”,4,0);
```

}

The this() keyword can be invoked only from a constructor and should be the first statement in the constructor.

State the significance of public, private, protected, default modifiers both singly and in combination.

Access specifiers are the attributes that determine whether a class member is accessible or not. Different types of access modifiers are as follows:

The public class is visible in other packages; the field is visible everywhere (the class must be public too)

The private methods or variables canbe used only by an instance of the same class that declares them. A private member can only be accessed by the class that declares the member.

The protected class is available to all classes in the same package and available to all subclasses of the class. This access is provided even to subclasses that reside in a different package from the class.

The default class members are accessible only to the class and other classes within that package.

The following table lists the different access specifiers in Java language and their impact on accessibility within a class, derived class, package, and external package:

Modifiers	Within Same Class	Within other class of Same package	Within derived class of other package	Within external Class of other package
Private (Class level A.S)	Yes	No	No	No
Default (Package level A.S)	Yes	Yes	No	No
Protected (Derived level A.S)	Yes	Yes	Yes	No
Public (Universal A.S)	Yes	Yes	Yes	Yes
A.S --> Access Specifier				

Figure 8.5: Access modifiers and accessibility

Explain different Java language keywords.

Final keyword: If a variable is declared as final, it's valued can't be changed. If a method is declared as final, it cannot be overridden in the subclass. A class declared as final cannot be inherited.

Static keyword: If we declare a method or a variable as static, it means that the data member belongs to the class rather than any specific instance of a class. Static methods and data members can be invoked without instantiating the class.

Abstract keyword: The abstract keyword is utilized to declare a class when we want to expose a certain functionality to the users and hide the underlying implementation. Youcannot create an instance of the abstract class. An abstract class can contains non-abstract and abstract methods.

Synchronized keyword: It controls the access to a code block in a multi-threaded Java application.

Native keyword: The native keyword is leveraged only for methods. This signals the compiler that the method has been coded in a language other than Java and signifies that the method lies outside the scope of the JRE.

What do you understand by a transient variable?

A transient variable is a variable that won't be serialized. This means that the JVM understands that the transient variable is not part of the persistent state of the object. For example:

```
class A implements Serializable {

  transient int i; // will not persist

  int j; // will persist

}
```

When an object of type A is written to persistent storage, the contents of i would not be saved, but the contents of j would be saved.

What is static in Java?

Static means one per class, not one for each object or instance no matter how many instances of a class exist. This means that youcan use them without creating an instance. Static methods are implicitly final because overriding is done based on the type of the object, and static methods are attached to a class, not an object. A static method in a superclass can be shadowed by another static method in a subclass, as long as the original method was not declared as final. Youcan't change a static method into an instance method in a subclass.

What do you understand by this() and super() in the context of constructors?

The this() is leveraged to invoke a constructor of the same class where as super() is leveraged to invoke a superclass constructor.

Classes, interfaces, constructors, and destructors

What is a final class?

A final class is a call that cannot be extended, that is, the final class may not be subclassed. A final method can't be overridden when its class is inherited. You can't change the value of a final variable and it is a constant.

What do you understand by boxing and unboxing?

Converting a primitive type like int, float, double, and so on to its corresponding wrapper class object (Integer, Float , and so on.) is known as boxing or autoboxing.

For example:

```
int i = 0;

Integer O = new Integer(i);
```

Unboxing is the opposite of boxing where the wrapper class is converted to a primitive type. For example:

```
i = O.intValue();
```

The following diagram depicts the boxing and unboxing process in Java:

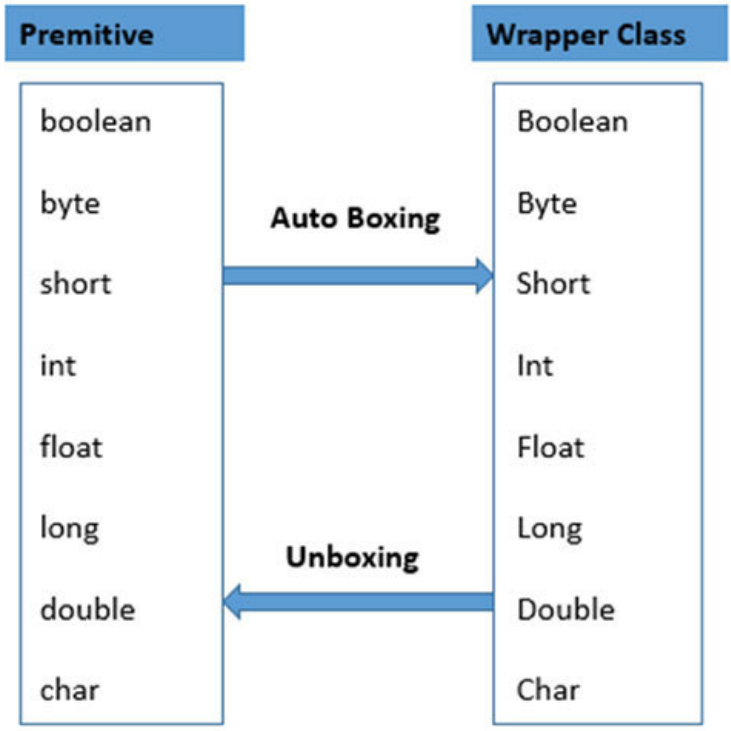


Figure 8.6: Boxing Vs Unboxing

How does Java distinguish between primitive types and objects?

Java distinguishes between primitive types and objects as follows.

Object classes (Integer, Double) have more methods. For example, Integer.toString() .

Primitive types, that is, double and int are compact and support arithmetic operators.

You need a wrapper to use object methods:

Integer ii = new Integer(i);

ii.hashCode()

Similarly, youneed to unwrap an object to do a primitive operation:

int j = ii.intValue() * 7;

Java makes this automatic:

ArrayList list = new ArrayList();

list.add(0, 42);

int total = list.get(0);

What is a pointer and does Java support pointers?

A pointer is a reference to a memory location. Improper handling of pointers leads to memory leaks and reliability challenges; hence, Java doesn't support pointers.

When does the compiler supply a default constructor for a class?

The compiler supplies a default constructor for a class if no other constructors are defined for that class.

What is the difference between an inner and nested class?

When a class is defined in the scope of another class, it becomes an inner class. If the access specifier of the inner class is static, then it becomes

a nested class.

If a class is declared without any access specifier, where can the class be accessed?

When a class is declared without an access specifier, it is said to have a package access. This means the class can only be accessed by other interfaces and classes that are defined within the same package.

What is a native method?

A native method is a method implemented in a language other than Java, that is, a language like C++.

What is an abstract method?

An abstract method is a method whose implementation is deferred to a subclass.

What is an object's lock?

An object's lock is a mechanism leveraged by multiple threads to obtain synchronized access to the object. A thread may execute a synchronized method of an object only after it has acquired the lock. All objects and classes have locks.

What is casting and down casting?

There are two types of casting: casting between primitive types and casting between object references. Casting between primitive types is leveraged to convert larger values such as double values to smaller values such as byte values. Casting between object references is leveraged to refer to an object by a compatible class, interface, or array type reference. Down casting is the casting from a general to a more specific type, that is, casting down the hierarchy. The following diagram depicts the casting and down casting process in Java:

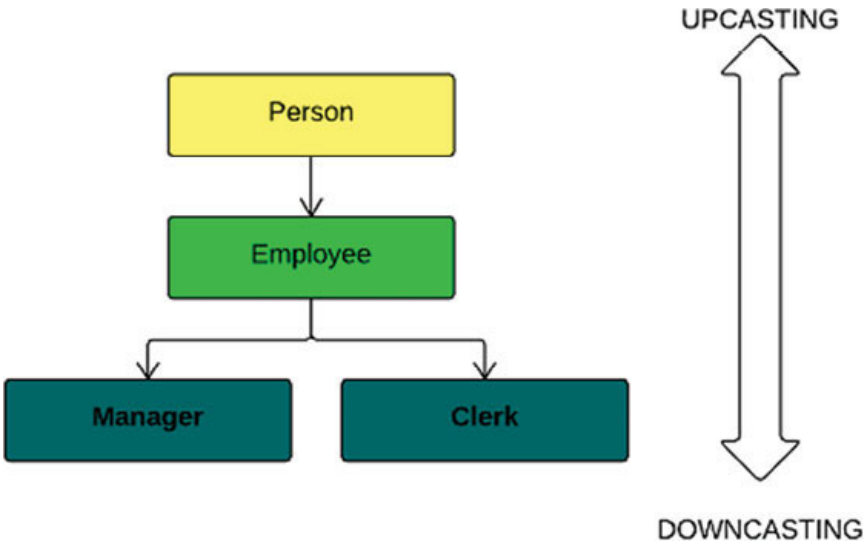


Figure 8.7: Casting vs down casting

Why does the Java framework not support operator overloading?

Operator overloading makes the code very difficult to manage and maintain. To ensure code simplicity, Java doesn't support the operator overloading feature.

Can we define private and protected modifiers for variables in interfaces?

No, only public and abstract modifiers are allowed for methods in interfaces.

What are local, member and class variables?

Variables declared within a method are called local variables. Variables declared within the class, that is, not within any methods are called member variables, also known as global variables. Variables declared within the class, that is, not within any methods are defined as static and they are called class variables.

Can we declare a method inside an interface as final?

This is not possible and ifit is done, it will result in compilation error(s):

The public and abstract are only specifiers for method declaration in an interface.

The interfaces do not provide implementation of methods and hence, an interface cannot implement another interface.

The interface can inherit from another interface; in fact, an interface can extend more than one interfaces.

Only public and abstract modifiers are allowed for methods in interfaces.

Why an interface is able to extend more than one interface but a class can't extend more than one class?

A class can extend only one class, but it can implement any number of interfaces. Java language doesn't allow multiple inheritance, so a class is restricted to extend only one class. But an interface is a pure abstraction and doesn't have inheritance hierarchy like classes. So, an interface is allowed to extend more than one interface.

Can a class be defined inside an interface? Can an interface be defined inside a class?

Yes, a class can be defined inside an interface. Yes, an interface can be defined inside a class.

Inheritance and polymorphism

Explain method overriding.

Method overriding is defined as creating a non-static method in the subclass that has the same signature and return type as a method defined in the super class. The signature of a method includes the name of the method and number, type of arguments, and sequence.

```
public class Person {
  private String name;
  private String ssn;
  public void setName(String name) {
    this.name = name;
  }
  public String getName() {
    return name;
  }
  public String getId() {
    return ssn;
  }
```

The getId method is overridden in a subclass:

```
public class Employee extends Person {
  private String empId;
  public void setId(String empId){
    this.empId=empId;
  }
  public String getId() {
    return empId;
  }
```

```
}
```

What is the use of the super keyword?

It allows you to access methods and properties of the parent/base class. For example:

```
public class Person {
  private String name;
  public Person(String name) {
    this.name = name;
  }
  public String getName() {
    return name;
  }
}
public class Employee extends Person {
  private String empID;
  public Employee(String name) {
    super(name); // Calls Person constructor
  }
  public void setID(String empID){
  this.empID=empID;
  }
  public String getID(){
    return empID;
  }
}
```

What are some alternatives to inheritance?

Delegation is an alternative to inheritance. Delegation means that you can include an instance of another class as an instance variable and forward messages to the instance. It is often safer than inheritance because it forces you to think about each message you forward because the instance is of a known class, rather than a new class. Because it doesn't force you to accept all the methods of the super class, you can provide only the methods that really make sense. On the other hand, it makes you write more code, and it is harder to reuse (because it is not a subclass).

Does Java support multiple inheritances?

Java doesn't support multiple inheritance. Java leverages primitive data types and hence, it is not a pure object-oriented language.

What is difference between method overriding and overloading? What restrictions are placed on method overriding?

Overriding is a method with the same name and arguments as the parent, whereas overloading is a method with the same method name but different arguments. Overridden methods must have the same name, argument list, and return types.

The overriding method may not limit the access of the method it overrides. The overriding method may not throw any exceptions that may not be thrown by the overridden method. A method cannot be overridden based on different return types and same arguments, as the methods can be called without leveraging their return type in which case it will be ambiguous for the compiler.

Does a class inherit the constructors of its superclass?

A class does not inherit constructors from its superclasses.

Which object-oriented concept is realized by leveraging overriding and overloading?

Polymorphism is realized through OOPs concepts of overriding and overloading

How to prevent a class from being inherited?

Declaring the class as final prevents it from being inherited by subclass(es). You can't define a class as final if it is already an abstract class. A class declared as final can't be extended by other classes.

Can a class be declared as protected?

A protected specifier cannot be leveraged for classes and interfaces. Fields and methods can be declared as protected in a class. However, fields and methods in an interface can’t be declared as protected. A protected method can be accessed by classes within the same package or by the subclasses of the class in other packages.

Miscellaneous

What is a Java package?

A Java package is a naming context for classes and interfaces. A package is leveraged to create a separate namespace for groups of classes and interfaces. Packages are also leveraged to organize related classes and interfaces into a single unit and to control accessibility to these classes and interfaces.

The package is a collection of related classes and interfaces. The first statement in a Java class is the package declaration. The java.lang packageis imported by default without a package declaration.

What are the uses of Java packages?

To organize related set of classes and interfaces, a package is leveraged. The package also defines the scope of a class. An application consists of several thousand classes and interfaces, and therefore, it is very critical to organize them logically into packages. By definition, the package is a group of related types that enable namespace management and access protection.

What is the significance of the import keyword?

To access variables and methods of a class Car in another class Person , we need to include the class Car in class Person . This is done using the import keyword. The import keyword followed by a fully qualified class name needs to be placed before the application class definition.

```
import classname ;

 // Class Definition.

import Java.util.Date;

import Java.io.*;

// Class Definition
```

The preceding code tells the JRE to import all the classes in a package. All the classes in java.io package are included implicitly.

What do you understand by a user-defined package?

When you create a Java application, you create many classes. You can organize these classes by creating custom packages. These packages created are called user-defined packages. A user-defined package contains one or more classes that can be imported into any Java application.

```
package land.vehicle;

public class Car

{

  String brand;

  String color;
```

```
  int wheels;

}
```

How to compile a Java source file and run a compiled Java program?

Compiling: Go to the location where the Java file is located. Enter the javac command followed by the Java source file name. For example, to compile the person class, type the javac Person.java command. Class files will be generated in the same location.

“javac Person.java”

Running: Go to the location where the Java class file is located. Enter the java command followed by the class file. For example, to execute the Person class, the type command

“java Person.class”

What does the java.util package provide?

The java.util package provides different utility classes and interfaces for Java applications that support:

String manipulations and management

Date and calendar functions

Collections and data structures

How to add comments in Java applications?

The comments can be inserted in the following three ways:

```
// comment type1

/* multi-line comments

comment line 2 */

/* documentation comment

comment line 2 */
```

This comment will be added in the automatic document generation process.

What is the difference between an error and an exception?

Runtime exceptions are those exceptions that are thrown at runtime because of either wrong input data or because of wrong business logic, etc. These are not checked by the compiler at compile time. An error is an irrecoverable condition occurring at runtime such as the Out Of Memory error. You cannot repair JVM errors at runtime. While exceptions are conditions that occur because of bad input. For example, File Not Found Exception will be thrown if the specified file does not exist or a Null Pointer Exception will take place if you try using a null reference. In most of the cases, it is possible to recover from an exception; for example, by giving the user feedback for entering proper values.

What is the process for throwing an exception object of a class?

The class should extend from the Exception class. You can extend your class from some more precise exception types.

How does exception permeate through the code?

An unhandled exception moves up the method stack in search of a match. When an exception is thrown from a code, which is wrapped in a try block followed by one or more catch blocks, a search is made for matching the catch block. If a matching type is found, then that block will be invoked. If a matching type is not found, then the exception moves up the method stack and reaches the caller method.

The same procedure is repeated if the caller method is included in a try…catch block. This process continues until a catch block handling the appropriate type of exception is found. If it does not find such a block, then finally the program terminates.

What are the different ways to handle exceptions?

There are two ways to handle exceptions:

By wrapping the desired code in a try block followed by a catch block to catch the exceptions.

List the desired exceptions in the throws clause of the method and let the caller of the method handle those exceptions.

Is it necessary that each try block must be followed by a catch block?

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block or a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

If I write return at the end of the try block, will the finally block still execute?

Yes, even if you write return as the last statement in the try block and no exception occurs, the finally block will execute. The finally block will execute and then the control returns.

Does importing a package import the subpackages as well? Example: Does import com.MyTest.* also import com.MyTest.UnitTests.*?

No, you will have to import the subpackages explicitly. For example, importing com.MyTest.* will import classes in the MyTest only package. It will not import any class in any of its subpackages.

What is the StringBuffer class?

StringBuffer is used to create and manipulate strings, which undergoes frequent modifications. If we define a StringBuffer class without specifying the data it can hold, it will allocate 16 bytes. The advantage of the string buffer is that if more data is added to buffer than the allotted size, its capacity will grow dynamically to handle additional characters. The + and += operators can't be used for string concatenation. It provides few utility methods, for example, trim substring.

What is the difference between the StringBuffer and StringBuilder class?

The StringBuffer class is not threadsafe, whereas StringBuffer is thread safe. It means that the methods of the StringBuffer class are synchronized. Compared to StringBuffer , StringBuilder is an improvised version. It allows quick string concatenation. It is not safe to be leveraged in multiple threaded applications since the methods are not synchronized. Creating new strings in a loop will be inefficient because of the huge number of new objects that are created and discarded.

StringBuilder.append()

StringBuilder.insert()

The following table provides a comparison between StringBuffer and StringBuilder classes:

Factor / Class	String	StringBuffer	StringBuilder
Mutability	Immutable	Mutable	Mutable
Thread Safety	Not thread safe	Thread safe	Not thread safe
Performance	Very high	Moderate	Very high

Figure 8.8: Difference between String, StringBuffer and StringBuilder classes

Describe synchronization with respect to multi-threading.

Synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared variable while another thread is in the process of updating or leveraging the same shared variable. This usually leads to errors or exceptions conditions.

Explain different ways of using the thread.

The thread could be implemented by leveraging a runnable interface or by inheriting the thread class. The latter is more advantageous, as when you are going for multiple inheritance, only an interface can help.

What is the difference between while{} and do while{} loops?

A while statement checks the condition at the beginning of a loop to see if the next iteration should be executed. A do…while statement checks the condition at the end of a loop to see if the next iteration of a loop should be executed. The do…while statement will always execute the body of a loop at least once.

What is the difference between prefix and postfix forms of the ++ operator?

The prefix operator performs the increment operation and returns the value of the incremented operation. The postfix operator returns the current value of the expression and then performs the increment operation.

What is the order of precedence and associativity?

Order of precedence defines the order in which operators get evaluated in the expressions. Precedence determines whether an expression is evaluated from left to right or right to left. The following table depicts the operator precedence and associativity:

Operators	Associativity	Type
++ --	right to left	unary postfix
++ -- + - ! (*type*)	right to left	unary prefix
* / %	left to right	multiplicative
+ -	left to right	additive
< <= > >=	left to right	relational
== !=	left to right	equality
&	left to right	boolean logical AND
^	left to right	boolean logical exclusive OR
|	left to right	boolean logical inclusive OR
&&	left to right	conditional AND
||	left to right	conditional OR
?:	right to left	conditional
= += -= *= /= %=	right to left	assignment

Figure 8.9: Order of precedence and associativity in Java

How is the ternary operator written x : y ? z or x ? y : z ?

The ternary operator is written as x ? y : z.

What is the purpose of declaring a final variable?

A final variable's value is constant and can't be changed. The final variables need to be initialized before using them. A method declared as final cannot be overridden. A subclass cannot have the same method signature with a different implementation.

Conclusion

This chapter was written to help you understand the basic concepts of Java programming for interview purposes. All the important Java concepts are explained with examples for your easy understanding. Java is used by approximately 10 million developers worldwide to develop applications for billion devices and applications, including the Big Data analytics.

This chapter presented an exhaustive question bank for Java programming language with special emphasis on practical scenarios and business cases. It covered aspects pertaining to Java iterator, classes, exceptions, JVM, JRE, collections, serialization, and polymorphism.

The next chapter will cover the database management system domain. The chapter will cover aspects related to core concepts, normalization, CRUD commands, joints, views, stored procedures, and triggers.

CHAPTER 9

Database Management System

Adatabase management system(DBMS) is a software application that interacts with the enduser, other applications, and the database to manage, analyze and store data. A DBMS is designed to allow the definition, creation, querying, updation, and administration of data in databases. Well-known DBMSs include Microsoft SQL, MongoDB, MySQL, Oracle, Sybase, and IBM DB2. A DBMS can interoperate by leveraging standards such as ODBC, JDBC, and SQL to allow a single application to work with more than one DBMS. DBMSs are often classified according to the database

models they support; the most popular database management systems support the relational model as represented by the SQL language. A database is a structured collection of data, schemas, tables, queries, reports, views, procedures, and other database objects.

This chapter covers the DBMS domain. It includes a questions bank for various aspects of the DBMS. It consists of different sections that cover core concepts, keys and constrains, SQL commands, normalization, stored procedures, functions, triggers, views, joins and miscellaneous topics.

Concepts

What is a DBMS?

A DBMS is a collection of programs that enables you to store, retrieve, update and delete information from a database. A DBMS is a collection of programs that enables management of data. In other words, it is a software application that provides the users with the processes of defining, constructing and manipulating data for various applications. A database is a logically coherent collection of data with some inherent meaning, representing some aspects of the realworld and it is designed, built and populated with data for a specific purpose.

The DBMS provides ways to organize, store, retrieve, and interact with the data. DBMS consists of:

A modeling language, which is leveraged to define database schema, or structure. Common database models include hierarchical, relational, network and object-based. These models differ in how they connect related information. The most commonly leveraged, particularly in web applications, is the relational database model.

A database engine that manages the structure and optimizes the storage, whether that is records, fields, objects or files for a balance between quick retrieval and efficient usage of storage space.

A transaction mechanism that validates data against allowed types before storing them in the DBMS and ensures that multiple users cannot update the same data simultaneously, thus potentially avoiding the corruption the data.

A database query language like SQL that enables you to write programs that extract data from the database, presents it to the enduser, and save and store updates.

The following diagram depicts a typical the three-tier architecture:

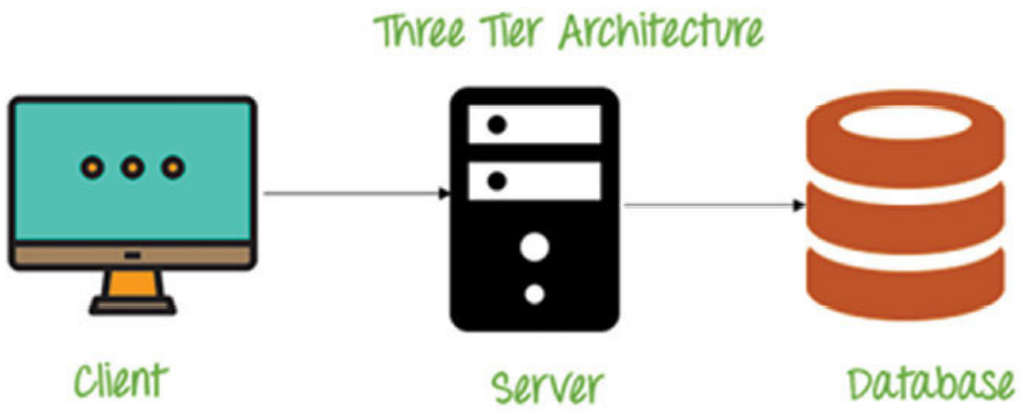


Figure 9.1: Typical three-tier architecture

A DBMS enables the creation of the database, updates it with data and creates ways to query and change that data without having to bother about technical aspects of storage and retrieval. The features of DBMS include the following:

User access and security management providing appropriate data access to multiple users while also protecting sensitive data.

Data backup to ensure high-availability of data.

Access logs, making it easier to see how the database is being leveraged.

Rules to ensure only data of the prescribed type is stored in each database field, for example, date fields may be set to contain a set range of dates.

Formulas such as summing, averaging, and counting for making statistical analysis and representation of the data simpler.

Performance monitoring and optimization tools to allow tuning the database settings for efficiency and speed.

The following diagram depicts the key characteristics of a DBMS:

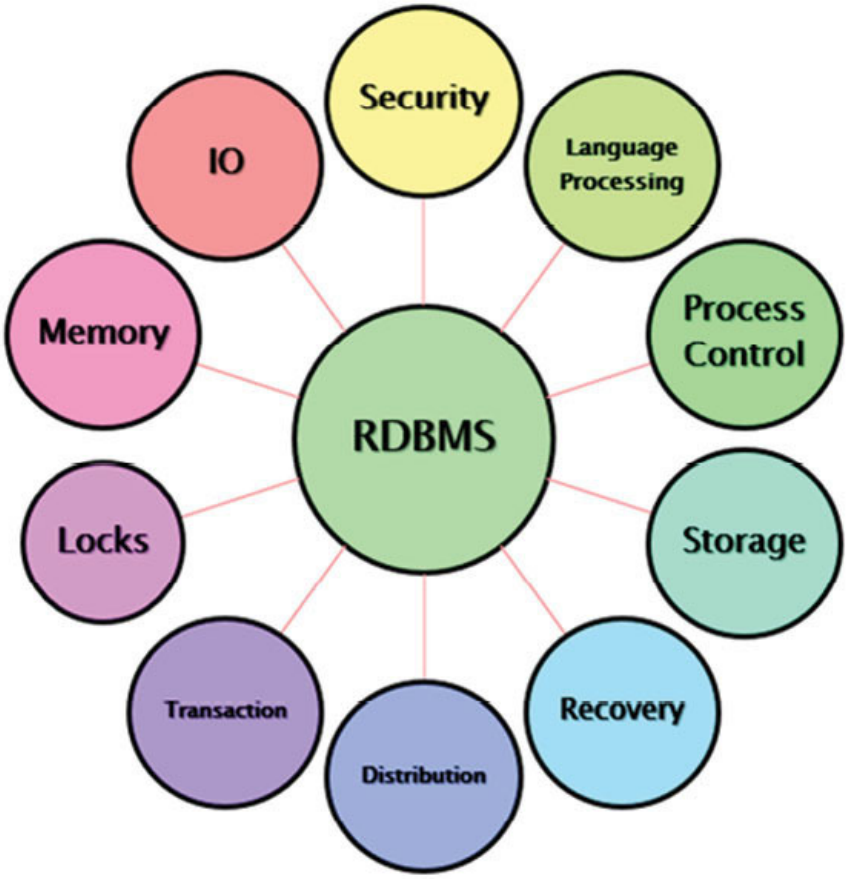


Figure 9.2: Key aspects of RDBMS

The advantages of DBMS are as follows:

Redundancy is controlled

Unauthorized access is controlled

Enforcing referential integrity constraints

Facilitating backup and recovery

The disadvantages in the file processing system are as follows:

Data redundancy and inconsistency

Difficulty in accessing data

Data isolation and integrity

Security issues

Concurrent access to files is not possible

What is RDBMS?

A relational database management system (RDBMS) is a database management system that is based on the relational model. Data from the relational database can be accessed or re-assembled in many ways without having to re-organize the database tables. Data from the relational database can be accessed using Structured Query Language (SQL) .

What is SQL? What are the different types of SQL statements?

SQL is a language designed for communicating with databases. SQL is an American National Standards Institute (ANSI) standard. SQL is a non-procedural language that is designed for data access operations on a normalized RDBMS. The primary difference between SQL and other

conventional programming languages is that SQL statements specify what data operations should be performed rather than how to perform them.

SQL statements are broadly classified into the following three:

Data Definition Language (DDL): DDL is leveraged to define the structure that holds the data.

GRANT: To provide user access

DENY: To deny permissions to users

REVOKE: To remove user access

Data Manipulation Language (DML): DML is leveraged for manipulation of the data. Typical operations are insert, update, delete, and retrieval. The Select statement is a limited version of the DML since it can't change the data in the database, but it can perform operations on data before the results are returned to the end user.

SELECT: To select specific data from a database

INSERT: To insert new records into a table

UPDATE: To update existing records

DELETE: To delete existing records from a table

Data Control Language (DCL): DCL is leveraged to control the visibility of data by granting access and setting privileges to create tables, and so on. For example, Grant and Revoke Manages the access permission to the enduser to access data.

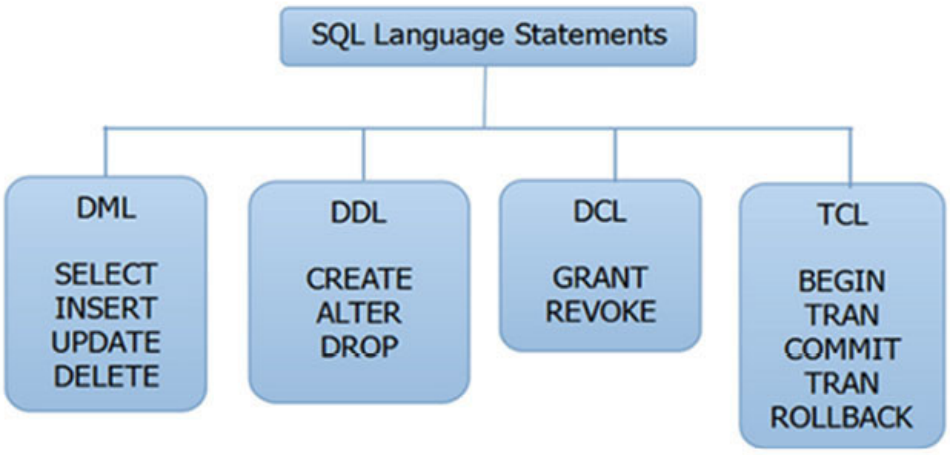


Figure 9.3: Different types of SQL statements

The advantages of SQL are as follows:

SQL is not a proprietary database programming language. Almost every major DBMS supports SQL and enables interaction with databases like MSSQL, Oracle, and MySQL.

SQL is easy to learn. The SQL statements are made up of descriptive English words.

SQL is a very powerful language and by leveraging its language elements, you can perform complex and sophisticated database operations.

What is a database transaction?

A database transaction takes the database from one consistent state to another. At the end of the transaction, the status should reflect the successful completion of the transaction or the system in the earlier state if the transaction fails.

What are properties of a transaction?

The properties of the transaction can be summarized as ACID properties:

Atomicity: A transaction consists of many steps. When all the steps in a transaction are complete, they will be reflected in the database, or if any of

the steps fail, all the transactions are rolled back.

Consistency: The database will step from one consistent state to another if the transaction succeeds and remains in the original state if the transaction fails.

Isolation: Every transaction should operate as if it is the only transaction in the application.

Durability: Once a transaction is successful, the updated records should be available for all other transactions on a permanent basis.

What is the purpose of acid properties?

ACID stands for Atomicity, Consistency, Isolation, and Durability and it plays an important role in the database.

These properties allow the database to be easily accessible and usable. This allows data to be shared more safely in between the tables.

If these properties are not being implemented, then the data will become inconsistent and inaccurate.

They help in maintaining the accuracy of the data in the database.

The following diagram depicts the ACID properties of the DBMS:

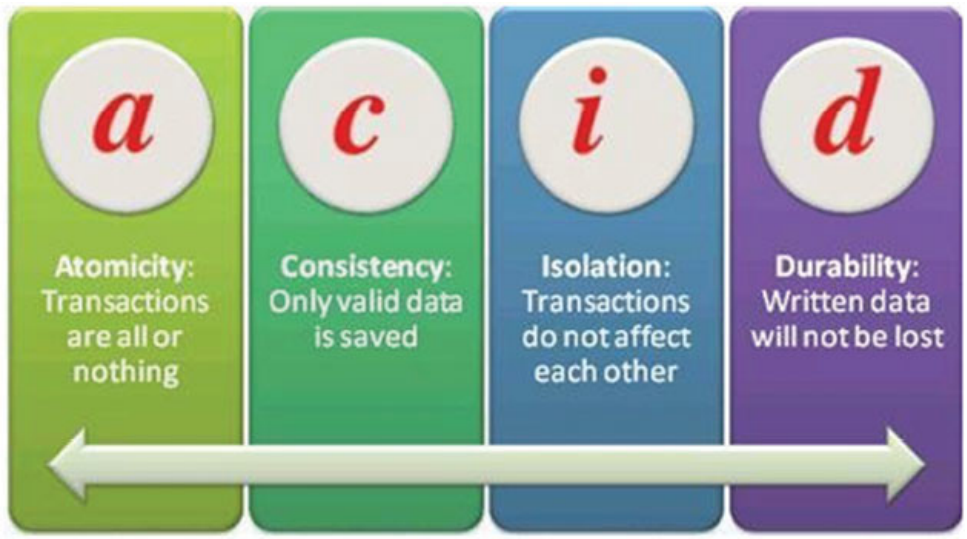


Figure 9.4: Core pillars of OOPs

What is a database lock? What are the different types of locks?

Locks enable locking the database records during a transaction and tell if the data item is currently being leveraged by other transactions in the application.

The different types of locks are as follows:

Shared lock: When a shared lock is applied to the data item, other transactions can only read the item, but can't write onto it.

Exclusive lock: When an exclusive lock is applied on the data item, other transactions can't read or write onto the data item.

What is data independence?

Data independence means that the application is independent of the storage structure and access mechanism. This means that the ability to modify the schema definition in one level should not affect the schema definition in the next higher level. There are two types of data independence:

Logical data independence: Modification in the logical level will affect the view level.

Physical data independence: Modification in the physical level should not affect the logical level.

What is a data model, an E-R model, and an object-oriented model?

Data model: It is a collection of conceptual tools for describing data, relationships, semantics, and constraints for a database.

E-R model: It is based on a real world that consists of basic objects called entities and object relationships. Entities are described in a database by a set of attributes.

Object-oriented model: It is based on a collection of objects. An object contains values stored in the variables and provides methods that operate on the object.

What is an entity, entity type, entity set, and extension of entity type?

An entity is a thing in the real world with an independent existence. An entity type is a collection (set) of entities that have the same attributes. An entity set is a collection of all entities of a particular entity type in the database. Extension of entity is a collection of entities of a particular entity type that are grouped together into an entity set.

What is normalization? What are the different types of normalization?

Normalization is a process of analyzing the given relation schemas based on their functional dependencies and primary keys to achieve the following properties:

Minimize redundancy

Minimize insertion, deletion, and update anomalies

There are different types of normalization. In the database design, one starts with one single table, with all possible columns. A lot of redundant data would be present since it's a single table. The process of removing the redundant data by splitting up the table in a well-defined manner is called normalization. The following are the different types of normalization:

First Normal Form (1NF): A relation is said to be in first normal form if all underlying domains contain only atomic values. After 1NF , we can still have redundant data.

Second Normal Form (2NF): A relation is said to be in 2NF if it is in 1NF and every non-key attribute is fully dependent on the primary key. After 2NF , we can still have redundant data.

Third Normal Form (3NF): A relation is said to be in 3NF , if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key.

Fourth Normal Form (4NF): A relation schema R is said to be in 4NF if for every multivalued dependency X Y that holds over R , one of following is true.

1.) X is a subset or equal to (or) XY = R.

2.) X is a super key.

Fifth Normal Form (5NF): A relation schema R is said to be 5NF if for every join dependency {R1, R2, …, Rn} that holds R . one the following is true:

1. Ri = R for some i .
2. The join dependency is implied by the set of FD, over R in which the left side is key of R .

The following diagram depicts the various database normal forms:

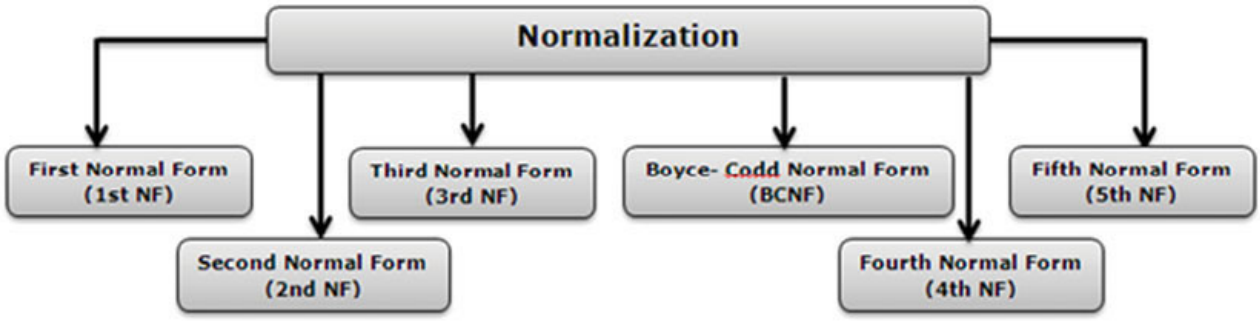


Figure 9.5: Types of normal forms

What is denormalization?

Denormalization is the process of boosting up database performance and adding of redundant data which helps to get rid of complex data. Denormalization is a part of database optimization technique. This process is used to avoid the use of complex and costly joins. Denormalization doesn't refer to the thought of not to normalize instead of that denormalization takes place after normalization. In this process, firstly the redundancy of the data will be removed using normalization process than through denormalization process we will add redundant data as per the requirement so that we can easily avoid the costly joins.

What is indexing? What are the different types of indexing?

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and the use of more storage space to maintain the extra copy of data.

Data can be stored only in one order on the disk. To support faster access according to different values, a faster search like a binary search for different values is desired. For this purpose, indexes are created on tables. These indexes need extra space on the disk, but they allow a faster search according to different frequently searched values. Indexing is a technique to ensure that the data can be quickly retrieved from the database tables. The different types of indexing are as follows:

B-tree indexing

Binary search style indexing

Inverted list indexing

Table indexing

Memory resident table

Name the subsystems of an RDBMS.

The RDBMS subsystems are as follows:

Memory management

I/O management

Security

Storage management

Language processing

Process control

Logging and recovery

Lock management

Distribution andtransaction control

What is the difference between a DBMS and RDBMS?

A DBMS provides a systematic and organized way of storing, managing and retrieving data from a collection of logically related information. An RDBMS also provides what DBMS provides but it provides relationship integrity:

A DBMS is persistent and accessible when the data is created or when it exists, but an RDBMS tells us about the relation between the table and other tables.

An RDBS supports a tabular structure for data and relationship between them in the system, whereas a DBMS supports only the tabular structure.

A DBMS provide uniform methods for applications that have to be independently accessed, but an RDBMS doesn’t provide methods like a DBMS but providesa relationship, which links one entity to another.

Describe the three levels of data abstraction.

Physical level: This is the lowest level of abstraction that describes how data is stored.

Logical level: This is the next higher level of abstraction, and it describes what data is stored in the database and the relationship between the

data.

View level: This the highest level of abstraction that describes only part of the entire database.

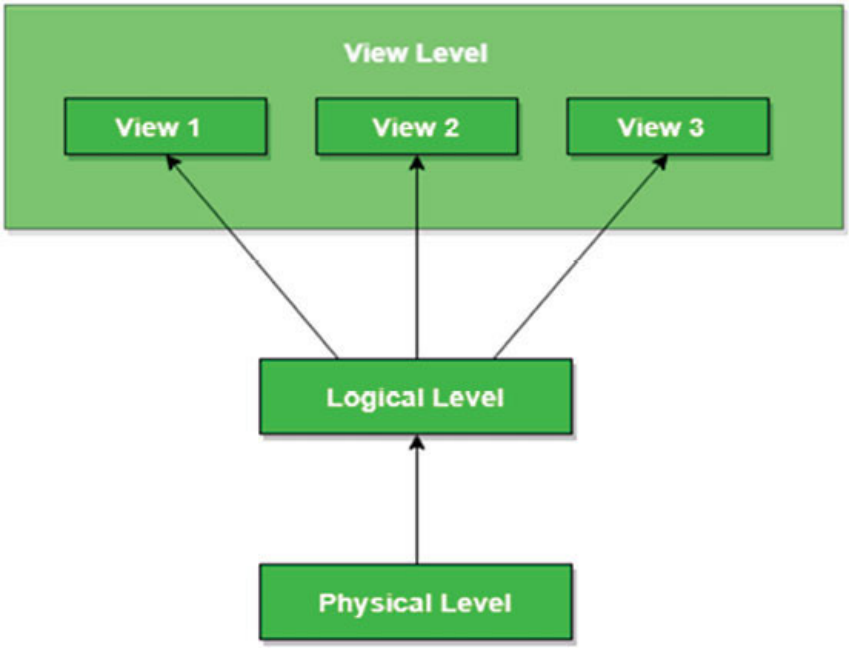


Figure 9.6: Levels of data abstraction

What do you understand by a relation in relational database model?

A relation in the relational database model is defined as a set of tuples that have the same attributes. A tuple represents an object and the information that the object contains. Objects are basically instances of classes and are used to hold the larger picture. Relation is described as a table and is organized in rows and columns. The data referenced by the relation comes in the same domain and has the same constraints as well. Relations in the relational database model can be modified using the commands like insert, delete, and so on.

The following diagram depicts the relational database model:

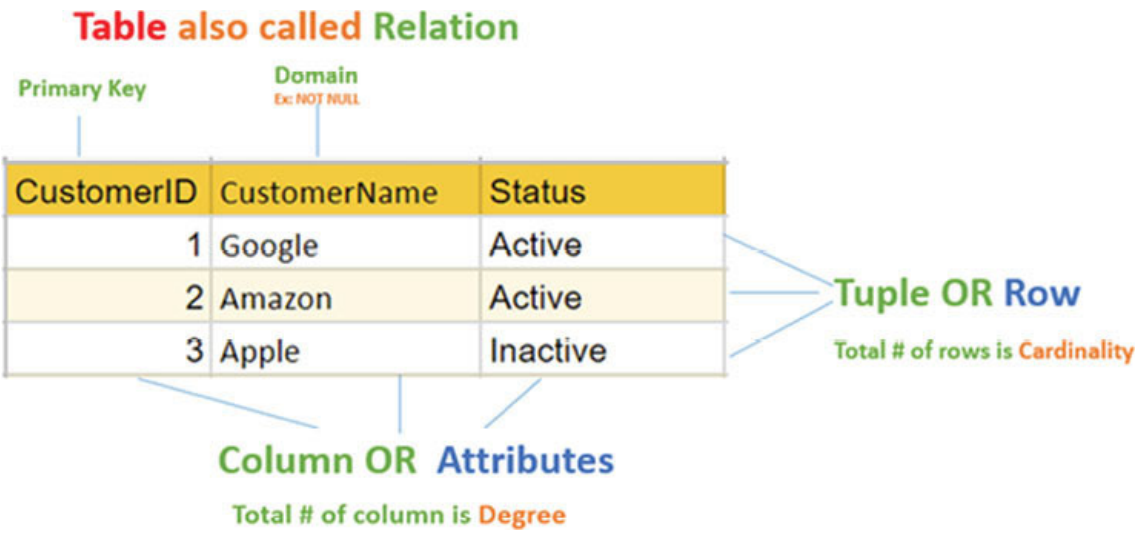


Figure 9.7: Relations in RDBMS

Keys and constraints

What is a primary key, composite primary key, foreign key, and unique key?

Primary key: A primary key is a column whose values uniquely identify each row in a database table. Primary key values can never be reused. If a row is deleted from the table, its primary key may not be assigned to any other or new rows in the future. To define a field as a primary key, the following conditions have to be met:

No two rows can have the same primary key value.

Every row must have a primary key value.

The primary key field cannot be NULL.

The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.

Composite primary key: A composite primary key is a set of columns whose values uniquely identify every row in a table. A table that contains a composite primary key will be indexed based on the columns specified in the primary key. This key will be referred in foreign key tables. For example, if the combined effect of columns such as Employee_ID and Employee Name in a table is required to identify a row uniquely, it's called a composite primary key . In this case, both the columns will be represented as the primary keys.

Foreign key: When a one table's primary key field is added to a related many table in order to create the common field, which relates the two tables, it is called a foreign key in the many tables. For example, the salary of an employee is stored in the salary table. The relation is established via a foreign key column Employee*ID*Ref , which refers to the Employee_ID field in the Employee table.

Unique key: A primary key is used to identify a row (record) in a table, whereas a unique key is used to prevent duplicate values in a column with the exception of a null entry.

The following diagram depicts the various keys and constraints in an RDBMS:

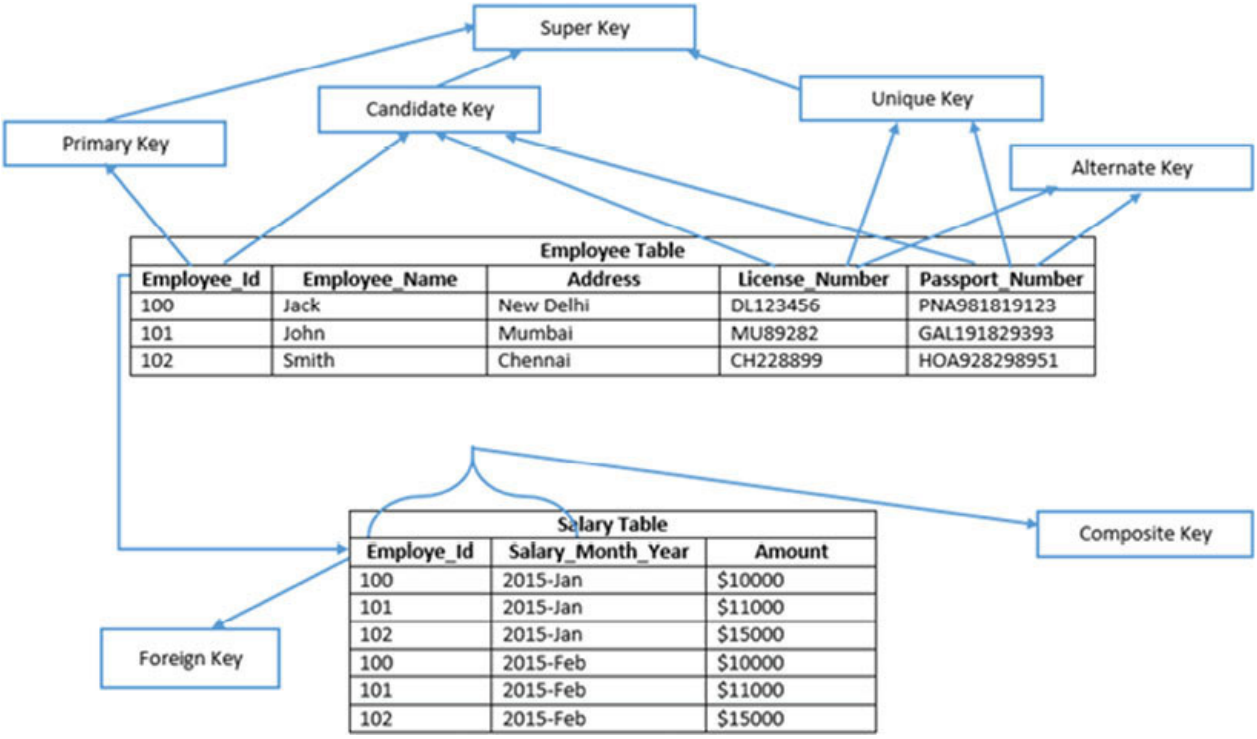


Figure 9.8: Types of keys in RDBMS

Define the integrity rules.

There are two integrity rules:

Referential integrity: It states that a foreign key can be either a NULL value or a primary key value of another relation.

Entity integrity: It states that a primary key cannot have a NULL value.

What is a query?

A query in a DBMS relates to user commands that are leveraged to interact with a database. The query language can be classified into DDL and DML.

What are the different types of relationships in the DBMS?

A relationship in the DBMS depicts an association between the tables. The different types of relationships are as follows:

One-to-One: This relationship states that there should be a one-to-one relationship between the tables i.e. there should be one record in both the tables. For example, among a married couple, both wife and husband can have only one spouse.

One-to-Many: This relationship states that there can be many relationships for one, that is, a primary key table can hold only one record, which can have many; one or none records in the related table. For example, a mother can have many children.

Many-to-Many: This relationship states that both the tables can be related to many other tables. For example, one can have many siblings and so do they have.

The following diagram depicts the various relationships in a RDBMS:

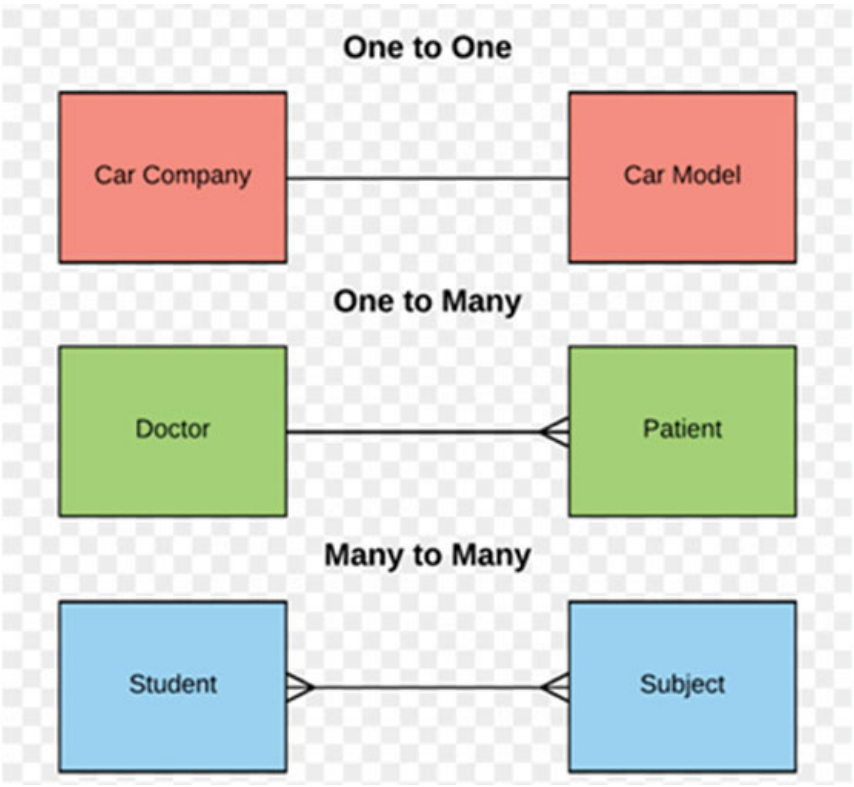


Figure 9.9: Types of relationships in RDBMS

What are constraints in a database?

Constraints are kinds of restrictions that are applied to the database or the domain of an attribute. For example, an integer attribute is restricted from 1-10 and not more than that. They provide the way to implement the business logic and the rules in a database. In a database, it can be implemented in the form of check constraints that check for the rules that haven't been followed by the programmer. Constraints are also used to restrict the data that can be stored in the relations. A domain constraint can be applied to check the domain functionality and keep it safe. Examples of constraints are:

Not null

Primary key

Foreign key

Unique key

Check

Default

SQL commands: insert, update, and delete

Define SQL insert, update and delete statements.

SQL insert statement: The SQL INSERT statement is leveraged to add rows to a table. For a row insert, the SQL query should start with insert into

statement followed by the table name and values command, followed by the values that need to be inserted into the table.

SQL update statement: SQL update is leveraged to update data in a row or set of rows specified in the filter condition. The format of an SQL UPDATE statement is the update command followed by the table name to be updated and set command followed by column names and their new values followed by the filter condition that determines which rows should be updated.

SQL delete statement: SQL delete is leveraged to delete a row or set of rows specified in the filter condition. The format of an SQL DELETE statement is the DELETE FROM command followed by the table name followed by the filter condition that determines which rows should be updated.

What are wild cards for pattern matching in the DBMS?

SQL like operator is leveraged for pattern matching in the DBMS. SQL like command takes more time to process. So before leveraging likean operator, consider suggestions given below on when and where to leverage a wildcard search:

When using wildcards, do not leverage them at the beginning of the search pattern, unless absolutely required. Search patterns that begin with wild cards have the slowest performance.

Pay meticulous attention to the placement of the wildcard symbols, and if they are misplaced, they might not return the data intended.

What is a checkpoint?

A checkpoint is like a snapshot of the DBMS state. By taking checkpoints, the DBMS checkpoint will reduce the amount of efforts to restore and restart in the event DBMS crashes.

What are the different states of the transaction?

A transaction in a database can be in one of the following states:

Active: In this state, the transaction is being executed. This is the initial state of every transaction.

Partially committed: When a transaction executes its final operation, it is said to be in a partially committed state.

Failed: A transaction is said to be in a failed state if any of the checks made by the database recovery system fails. A failed transaction can no longer proceed further.

Aborted: If any of the checks fails and the transaction has reached a failed state, then the recovery manager rolls back all its write operations on the database to bring the database back to its original state where it was prior to the execution of the transaction. Transactions in this state are called aborted. The database recovery module can select one of the two operations after a transaction aborts:

Restart the transaction

Kill the transaction

Committed: If a transaction executes all its operations successfully, it is said to be committed. All its effects are now permanently established on the database system.

The following diagram depicts the different states of a transaction:

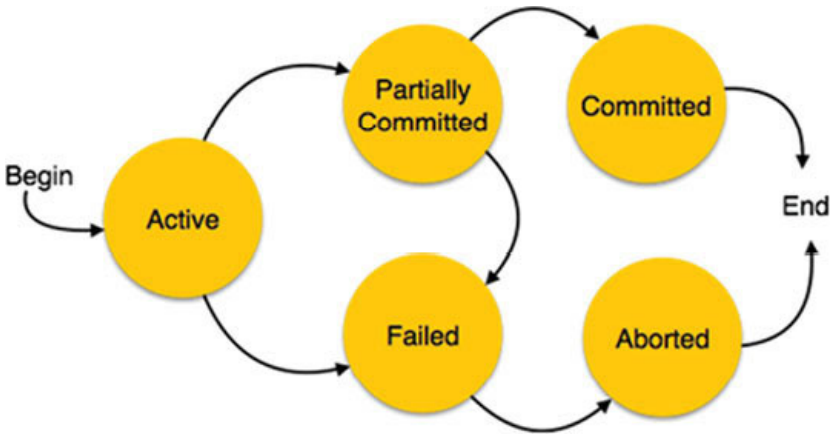


Figure 9.10: Different states of a transaction

Miscellaneous

Define a join and explain the different types of joins.

In order to avoid data duplication, data is stored in related tables. The join keyword is leveraged to fetch data from related tables. The join returns rows when there is at least one match in both the tables.

There are different types of joins, which are as follows:

Right join: It returns all the rows from the right table, even if there are no matches in the left table.

Left join: It returns all the rows from the left table, even if there are no matches in the right table.

Cross join: The cross join will return all the records where each row from the first table is combined with each row from the second table. The where clause functions like an inner join.

Full join: It returns rows when there is a match in one of the tables. This is like right and left joins at the same time.

Self join: The self join is a query leveraged to join a table to it. Aliases should be leveraged for the table comparison.

The following table describes the different types of joins in the DBMS:

Join Type	Purpose
Cross Join	Returns Cartesian product of the tables involved in the join
Inner Join	Returns only the matching rows. Non matching rows are eliminated.
Left Join	Returns all the matching rows + non matching rows from the left table
Right Join	Returns all the matching rows + non matching rows from the right table
Full Join	Returns all rows from both tables, including the non-matching rows.

Figure 9.11: Output of different types of SQL joins

What is a view? What are the advantages and disadvantages of views?

The views are virtual tables. Unlike tablets that contain data, views simply contain queries that dynamically retrieve data when leveraged. A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base tables. In other words, there is no stored file that directly represents the view instead a definition of the view is stored in the data dictionary. Growth and restructuring of base tables are not reflected in views. Thus, the view can insulate users from the effects of restructuring and growth in the database and hence, accounting for logical data independence.

The advantages of views are as follows:

Views don't store data at a physical location.

Views are leveraged to hide some of the columns from the table.

Views provide access restriction, since data insertion, updation, and deletion is not possible with the views.

The disadvantages of views are as follows:

When a table is dropped, attached views become irrelevant.

When a query requesting data from a view is triggered, it's a bit slow.

When views are created for large tables, it occupies more memory.

For example:

```
CREATE VIEW view_name AS

SELECT column_name(s)

FROM table_name

WHERE condition
```

What is a stored procedure?

A stored procedure is a function, which contains a collection of SQL queries. The procedure can take inputs, processes them and sends them back to the output data. A stored procedure is a named group of SQL statements that have been previously created and stored in the server database. Stored procedures accept input parameters so that a single procedure can be used over the network by several clients using different input data. When the procedure is modified, all the clients automatically get the new version. Stored procedures reduce network traffic and improve performance. Stored procedures can be used to help ensure the integrity of the database.

The advantages of stored procedures is that they are pre-compiled and stored in the database. This enables the database to execute the queries faster. Since many queries can be included in a stored procedure, a round trip time to execute multiple queries from the source to the database and back is avoided. The following is an example of a stored procedure:

```
CREATE Procedure Procedure_Name

(

  //Parameters

)

AS

BEGIN

SQL statements in stored procedures to update/retrieve records

END
```

The following diagram explains the stored procedure:

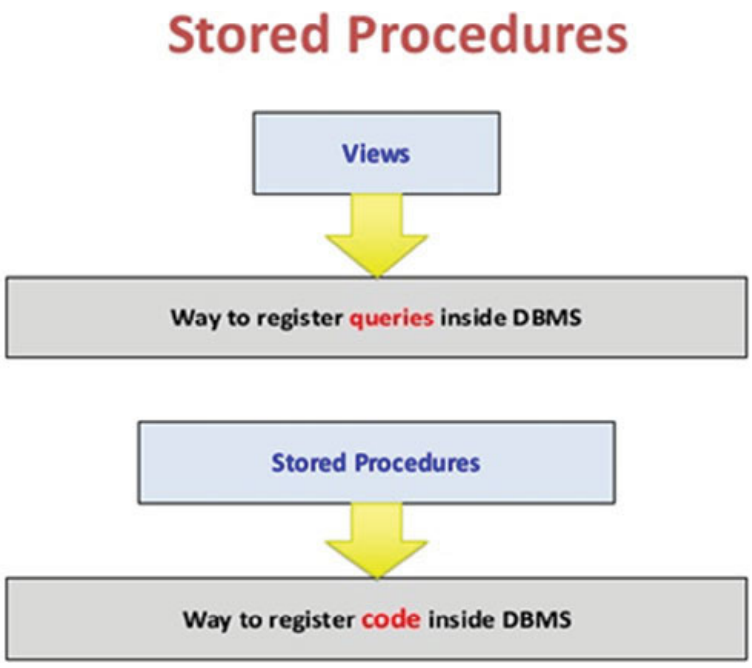


Figure 9.12: Stored procedures in DBMS

What is the difference between a stored procedure and function?

A procedure can have both input and output parameters, but a function can only have input parameters.

Inside a procedure, we can use the DML (INSERT/UPDATE/DELETE) statements. However, inside a function, we can't use the DML statements.

We can't utilize a stored procedure in a SELECT statement. However, we can use a function in a SELECT statement.

We can use a try…catch block in a stored procedure but inside a function, we can't use a try…catch block.

We can use transaction management in a procedure but we can't use it in a function.

We can't join a stored procedure but we can join functions.

Stored procedures cannot be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section. However, we can use a function anywhere.

A procedure can return 0 or n values (max 1024). But a function can return only 1 value that is mandatory.

A procedure can't be called from a function but we can call a function from a procedure.

Stored Procedure	Functions
Stored procedures can returns zero or N values.	Function can return single value which is compulsory.
Procedure can work with Insert, update, delete, select statements.	Functions only works with Select Statements
We can use functions inside the stored procedure.	Stored procedures can not be used inside the functions
Error handing is possible using try catch	Error handling not possible using try catch
Transaction are possible	Transactions are not possible
Stored procedure can be called by using exec followed by stored procedure name	Functions can be call by using the select statements followed by function name

Figure 9.13: Difference between stored procedures and functions

What is a trigger?

Database triggers are sets of commands that are executed when an event (before insert, after insert, on update, on delete of a row) occurs on a table and views.

A database trigger is a PL/SQL block that automatically executes when the insert, update, and delete statements are invoked on a table. The trigger can be defined to execute once for the entire statement or once for every row that is inserted, updated, or deleted. For any one table, you can define database triggers there are 12 events for which. A database trigger can call database procedures that are also written in PL/SQL.

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:

Database definition (DDL) statement (CREATE, ALTER, or DROP).

Database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Database manipulation (DML) statement (DELETE, INSERT, or UPDATE).

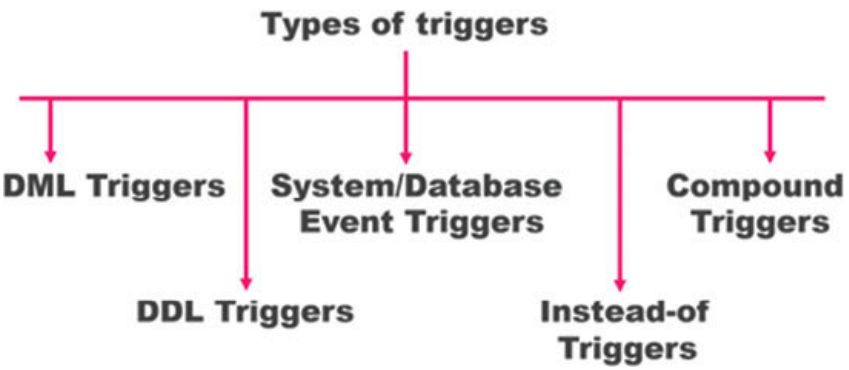


Figure 9.14: Different types of triggers

Explain the difference between DELETE, TRUNCATE, and DROP commands.

DELETE: This command deletes only the rows from the table based on the condition given in the where clause or deletes all the rows from the table if no condition is specified. But it does not free the space containing the table. The syntax of a SQL DELETE statement is as follows:

DELETE FROM table_name [WHERE condition];

TRUNCATE: This command is used to delete all the rows from the table and free the space containing the table. The syntax of a SQL TRUNCATE statement is as follows:

TRUNCATE TABLE table_name;

DROP: This command is used to remove an object from the database. If you drop a table, all the rows in the table are deleted and the table structure is removed from the database. Once a table is dropped, we cannot get it back, so be careful while using the DROP command. When a table is dropped, not all the references to the table will be valid. The syntax of a SQL DROP statement is as follows:

DROP TABLE table_name;

What is the difference between a cluster and non-cluster index?

A clustered index reorders the way records in the table are physically stored. There can be only one clustered index per table. It makes data retrieval faster.

A non-clustered index does not alter the way it was stored but creates a completely separate object within the table. As a result, the insert and update commands will be faster.

What are cursors?

A database cursor is a control that enables a traversal over the rows or records in the table. This can be viewed as a pointer to one row in a set of rows. A cursor is very much useful for traversing such as retrieval, addition and removal of database records. PL/SQL leverages cursors for all database information access statements. The language supports the use of the following two types of cursors:

Implicit

Explicit

What are the different types of database backups?

The different types of backups are as follows:

Copy-only backup: A specialuse backup that is independent of the regular sequence of SQL server backups.

Data backup: A backup of data in a complete database (a database backup), a partial database (a partial backup), or a set of data files or filegroups (a file backup).

Database backup: A backup of a database. Full database backups represent the whole database at the time the backup finished. Differential database backups contain only changes made to the database since its most recent full database backup.

Differential backup: A data backup that is based on the latest full backup of a complete or partial database or a set of data files or filegroups (the differential base) and that contains only the data extents that have changed since the differential base.

A differential partial backup records only the data extents that have changed in the filegroups since the previous partial backup known as the base for the differential.

Full backup: A data backup that contains all the data in a specific database or a set of filegroups or files and enough logs to allow you to recover that data.

Log backup: A backup of transaction logs that includes all the log records that were not backed up in a previous log backup. (full recovery model).

File backup: A backup of one or more database files or filegroups.

Partial backup: This contains data from only some of the filegroups in a database, including the data in the primary filegroup, every read/write filegroup, and any optionally specified read-only files.

What is database partitioning? Explain the importance of partitioning.

The division of a logical database into independent complete units for improving its management, availability and performance is called Database partitioning. Splitting of one table, which is large, into smaller database entities logically is called database partitioning. The benefits of database partitioning are as follows:

To improve query performance in situations dramatically when mostly rows, which are heavily accessed, are in one partition.

Accessing large parts of a single partition.

Slower and cheaper storage media can be used for data, which is seldom used.

The following diagram depicts the DBMS partitioning:

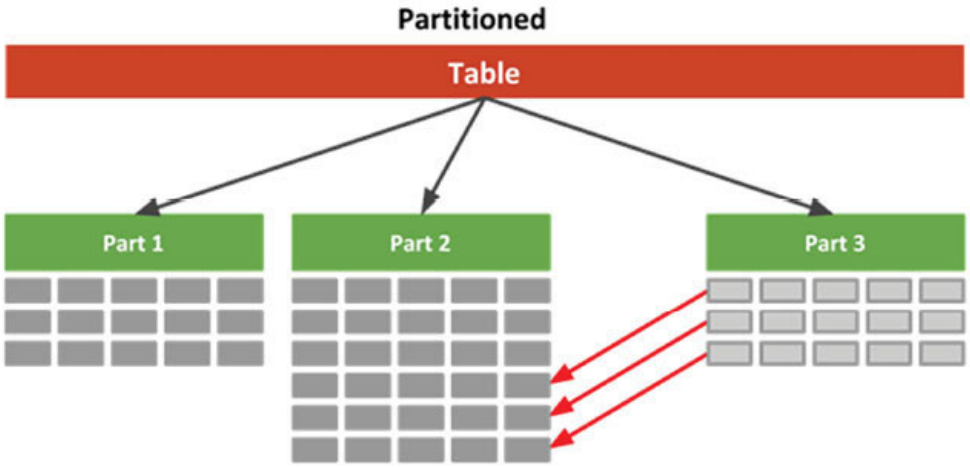


Figure 9.15: Database partitioning

What do you mean by atomicity and aggregation?

Atomicity: Atomicity states that database modifications must follow an all or nothing rule. Each transaction is said to be atomic. If one part of the transaction fails, the entire transaction fails.

Aggregation: A feature of the entity relationship model that allows a relationship set to participate in another relationship set. This is indicated on an ER diagram by drawing a dashed box around the aggregation.

Describe concurrency control.

Concurrency control is the process of managing simultaneous operations against a database so that the database integrity is not compromised. There are two approaches to concurrency control. The pessimistic approach involves locking and the optimistic approach involves versioning.

What is the difference between the WHERE clause and the HAVING clause?

The WHERE clause can be used with theselect, update, and delete statement clause but the HAVING clause can be used only with a select statement.

We can't use an aggregate functions in the WHERE clause unless it is in a subquery contained in the HAVING clause, whereas we can use an aggregate function in the HAVING clause. We can use a column name in the HAVING clause but the column must be contained in the group by the clause.

WHERE is used before the GROUP BY clause, whereas the HAVING clause is used to impose a condition on the GROUP function and is used after the GROUP BY clause in the query.

The WHERE clause applies to each and every row, whereas the HAVING clause applies to summarized rows (summarized with GROUP BY).

In the WHERE clause, the data that is fetched from memory depending on a condition, whereas in the HAVING clause, the complete data is first fetched and then separated depending on the condition.

How to select unique records from a table?

We can select unique records from a table using the DISTINCT keyword. The syntax of the DISTINCT keyword is as follows:

Select DISTINCT EmployeeId, EmployeeName from Employees.

Write an SQL SELECT statement to display all the columns of the STUDENT table but only those rows where the Grade column is greater than or equal to 90.

SELECT * FROM STUDENT WHERE Grade >= 90;

What are aggregate and scalar functions?

Aggregate functions are used to evaluate mathematical calculations and return single values. These can be calculated from the columns in a table. Examples of aggregate functions are as follows:

AVG(): This function returns the average value.

COUNT(): This function returns the number of rows.

MAX(): This function returns the largest value.

MIN(): This function returns the smallest value.

ROUND(): This function rounds a numeric field to the number of decimals specified.

SUM(): This function returns the sum.

Scalar functions return a single value based on the input value. Examples of scalar functions are UCASE() and NOW() , which arecalculated with respect to strings.

How to create a table?

For creating a table in a database, we need to use the CREATE TABLE statement. For example:

CREATE TABLE "SomeName" ("col1""datatype", "col2""datatype");

Define the SELECT INTO statement.

The SELECT INTO statement copies data from one table to a new table. The new table will be created with the column-names and types as defined in the old table. You can create new column names using the AS clause. The syntax for the AS clause is as follows:

SELECT * INTO newtable FROM oldtable WHERE condition;

Conclusion

A database management system (DBMS) is set of related software applications that provide end users and application programmers a systematic way to create and manage databases.

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

This chapter covered the DBMS domain. The chapter included a questions bank for various aspects of the DBMS, which includes core concepts, constrains, SQL commands, normalization, stored procedures, triggers, views, and joins.

The next chapter will cover question and answers for trending programming languages, which includes Python and R. It will present an exhaustive question bank with special emphasis on practical scenarios. It will cover topics such as programming constructs, pointers, memories, variables, OOPS, data abstraction, functions, and procedures.

CHAPTER 10

Trending Languages– Python and R Programming

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently, whereas other languages use punctuation, and it has a fewer syntactical constructions than other languages.

Python is a high-level object-oriented programming language. There are many benefits of using Python. Firstly, Python scripts are simple, shorter, portable and opensource. Secondly, Python variables are dynamic typed. So, you don’t need to think about the variable type while coding. Thirdly,

Python classes have no access modifiers, which Java has. So, you don't need to think about access modifiers. Lastly, Python provides us different libraries and data-structures to make our coding easier.

This chapter introduces you to the frequently asked questions on Python and R programming language interviews. This chapter is the perfect guidance for you to learn all about the vital aspects of trending programming languages. The chapter on Python Programming language covers core concepts, data types and structures, functions and miscellaneous vital programming aspects. The chapter on R programming covers core concepts, data types and structures, functions, packages and miscellaneous vital topics on programming aspects. This is one-stop resource from where you can boost your interview preparations.

Python programming

Core concepts

What are the key features of Python?

The key features of Python are as follows:

Python is an interpreted language. This means that, unlike languages like Cand its variants, Python does not need to be compiled before it is run. Other interpreted languages include PHP and Ruby.

Python is dynamically typed; this means that you don't need to state the types of variables when you declare them or anything like that. You can do things like x=111 and then x="I'm a string " without any errors.

Python is well suited to OOPs, which allows the definition of classes along with composition and inheritance. Python does not have access specifiers (like C++ public, private); the justification for this point is given, as we are all adults here .

In Python, functionsarefirstclass objects. This means that they can be assigned to variables, returned from other functions, and passed to functions. Classes are also first class objects.

Writing the Python code is quickbut running it is often slower than compiled languages. Fortunately, Python allows the inclusion of C-based extensions so bottlenecks can be optimized and often is. Thenumpy package is a good example of this. It's quite quick because a lot of the number crunching it does isn't actually done by Python.

Python can be usedin many spheres such as web applications, automation, scientific modeling, Big Data applications, and so on. It's also often used as glue code to get other languages and components to play nice.

What is the difference between deep and shallow copy?

Shallow copy is used when a new instance type is created and it keeps the values that are copied in the new instance. Shallow copy is used to copy the reference pointers just as if it copies the values. These references point to the original objects and the changes made in any member of the class will also affect the original copy. Shallow copy allows faster execution of the program and it depends on the size of the data that is used.

Deep copy is used to store the values that are copied. Deep copy doesn't copy the reference pointers to the objects. It makes a reference to an object and the new object that is pointed to other object values gets stored. The changes made in the original copy won't affect any other copy that uses the object. Deep copy makes the execution of the program slower due to making certain copies for each object that is been called.

What is the difference between lists and tuples?

How is multithreading achieved in Python?

Python has a multithreading package but if you want to multithread to speed up your code, then it's usually not a good idea to use it.

Python has a construct called the Global Interpreter Lock (GIL) . The GIL makes sure that only one of your threads can execute at any one time. A thread acquires the GIL, does a little work, and then passes the GIL to the next thread.

This happens very quickly so to the human eye it may seem like your threads are being executed in parallel, but they are just taking turns using the same CPU core.

All this GIL passing adds overhead to execution. This means that if you want to make your code run faster, then using the threading package often isn't a good idea.

How can the ternary operators be used in Python?

The ternary operator is an operator that is used to show the conditional statements. This consists of the true or false values with a statement that has to be evaluated for it. The syntax for the ternary operator is as follows:

[on*true] if [expression] else [on*false]x, y = 25, 50 big = x if x < y else y

For example:

The expression is evaluated like if x

How is memory managed in Python?

Memory management in Python is managed by the Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have access to this private heap. The Python interpreter takes care of this instead.

The allocation of the heap space for Python objects is done by Python's memory manager. The core API gives access to some tools for the programmer to code.

Python also has an inbuilt garbage collector, which recycles all the unused memory and so that it can be made available to the heap space.

Explain inheritance in Python.

Inheritance allows one class to gain all the members (say attributes and methods) of another class. Inheritance provides code reusability, makes it easier to create and maintain an application. The class from which we are inheriting is called a superclass and the class that is inherited is called a derived/child class.

Inheritance allows us to reuse our code and makes it easier to create and maintain applications. Python supports the following kinds of inheritance:

Single inheritance: A class inherits from a single base class.

Multiple inheritance: A class inherits from multiple base classes.

Multilevel inheritance: A class inherits from a base class, which, in turn, inherits from another base class.

Hierarchical inheritance: Multiple classes inherit from a single base class.

Hybrid inheritance: Hybrid inheritance is a combination of two or more types of inheritance.

Explain Flask and its benefits.

Flask is a web micro framework for Python based on Werkzeug, Jinja2 and good intentions. It's BSD licensed! Werkzeug and Jinja2 are two of its dependencies. Thismeans it will have little to no dependencies on external libraries. It makes the framework light while there is a little dependency to update and fewer security bugs.

A session allows you to remember information from one request to another. In flask, a session uses a signed cookie so the user can look at the session contents and modify. The user can modify the session if only it has the secret key, Flask.secret_key .

Whenever Python exits, why does not all the memory get deallocated?

Whenever Python exits, especially those Python modules, which are having circular references to other objects or the objects that are referenced from the global namespaces, are not always deallocated or freed.

It is impossible to deallocate those portions of memory that are reserved by the C library.

On exit, because of having its own efficient clean up mechanism, Python would try to deallocate/destroy every other object.

What is a dictionary in Python?

The built-in datatypes in Python is called a dictionary. It defines a one-to-one relationship between keys and values. Dictionaries contain pair of keys and their corresponding values. Dictionaries are indexed by keys.

Let's take a look at an example:

The following example contains some keys such as Country , Capital , and PM . Their corresponding values are India , Delhi , and Modi , respectively:

dict={'Country':'India','Capital':'Delhi','PM':'Modi'}

print dict[Country]

India

print dict[Capital]

Delhi

print dict[PM]

Modi

What is monkey patching in Python?

In Python, the term monkey patch only refers to dynamic modifications of a class or module at runtime.

Consider the following example:

m.py

class MyClass:

def f(self):

print "f()"

We can then run the monkey-patch testing as follows:

import m

def monkey_f(self):

print "monkey_f()"

m.MyClass.f = monkey_f

obj = m.MyClass()

obj.f()

The output will be as follows:

monkey_f()

As we can see, we did make some changes in the behavior off() in MyClass using the function we defined, monkey_f(), outside the modulem.

What do*args and*kwargs mean? And why would we use them?

We use *args when we aren't sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function.* kwargs is used when we don't know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments. The args and kwargs identifiers are conventions; you could also use *boband*billy but that would not be wise.

Write a one-liner that will count the number of capital letters in a file. Your code should work even if the file is too big to fit in memory.

with open(SOME*LARGE*FILE) as fh:

count = 0

text = fh.read()

for character in text:

    if character.isupper():

count += 1

We will now try to transform this into a single line code:

count sum (1 for line in fh for character in line if character.isupper())

What arenegative indexes and why are they used?

The sequences in Python are indexed and it consists of the positive as well as negative numbers. The numbers that are positive use 0 as the first index and 1 as the second index and the process goes on like this.

The index for the negative number starts from -1 that represents the last index in the sequence and -2 as the penultimate index and the sequence carries forward like the positive number.

The negative index is used to remove any new-line spaces from the string and allow the string to accept the last character that is given as S[:-1] . The negative index is also used to show the index to represent the string in the correct order.

How can you randomize the items of a list in place in Python?

Consider the following example:

from random import shuffle

x = ['Keep', 'The', 'Blue', 'Flag', 'Flying', 'High']

shuffle(x)

print(x)

The output of the following code is as follows:

['Flying', 'Keep', 'Blue', 'High', 'The', 'Flag']

What is the process of compilation and linking in Python?

The compiling and linking allows the new extensions to be compiled properly without any error and linking can be done only when it passes the compiled procedure. If the dynamic loading is used, then it depends on the style that is being provided with the system. The Python interpreter can be used to provide the dynamic loading of the configuration setup files and rebuild the interpreter. The steps that are required for compilation and linking are as follows:

Create a file with any name and in any language that is supported by the compiler of your system. For example, file.c or file.cpp .

Place this file in the modules/directory of the distribution, which is being used.

Add a line in the Setup.localfile that is present in the modules/directory.

Run the file using spam file.o .

After a successful run, rebuild the interpreter using the make command on the top-level directory.

If the file is changed, then run rebuild Make file by using the command as ‘make command .

Write a sorting algorithm for a numerical dataset in Python.

The following code can be used to sort a list in Python:

list = ["1", "4", "0", "6", "9"]

list = [int(i) for i in list]

list.sort()

print (list)

Look at the following code. Write down the final values of A0, A1, …An.

A0 = dict(zip(('a','b','c','d','e'),(1,2,3,4,5)))

A1 = range(10)A2 = sorted([i for i in A1 if i in A0])

A3 = sorted([A0[s] for s in A0])

```
A4 = [i for i in A1 if i in A3]

A5 = {i:i*i for i in A1}

A6 = [[i,i*i] for i in A1]

print(A0,A1,A2,A3,A4,A5,A6)
```

The following will be the final output of A0, A1, … A6:

```
A0 = {'a': 1, 'c': 3, 'b': 2, 'e': 5, 'd': 4} # the order may vary

A1 = range(0, 10)

A2 = []

A3 = [1, 2, 3, 4, 5]

A4 = [1, 2, 3, 4, 5]

A5 = {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}

A6 = [[0, 0], [1, 1], [2, 4], [3, 9], [4, 16], [5, 25], [6, 36], [7, 49], [8, 64], [9, 81]]
```

Explain the split(), sub(), and subn() methods of the “re” module in Python.

To modify the strings, Python’s re module provides the following three methods:

split(): It uses a regex pattern to split a given string into a list.

sub(): It finds all the substrings for which the regex pattern matches and then replaces them with a different string.

subn(): It is similar to sub() and also returns the new string along with the number of replacements.

How can you generate random numbers in Python?

The random module is the standard module that is used to generate the random numbers. The method is defined as follows:

```
import random

random.random
```

The random.random() method returns the floating point number that is in the range of [0, 1) . The function generates the random float numbers. The methods that are used with the random class are the bound methods of the hidden instances. The instances of the randommethod can be done to show the multithreading programs that create different instances of individual threads. The other random generators that are used are as follows:

randrange(a, b): It chooses an integer and defines the range inbetween [a, b) . It returns the elements by selecting them randomly from the range that is specified. It doesn’t build a range object.

uniform(a, b): It chooses a floating point number that is defined in the range of [a,b).lyt returns the floating point number.

normalvariate(mean, sdev): It is used for the normal distribution where the mu is a mean and the sdev is a sigma that is used for standard deviation.

The Random class that is used and instantiated creates independent multiple random number generators.

What is the difference between range and xrange?

For the most part, xrange and range are the same in terms of functionalities. They both provide a way to generate a list of integers for you to use; however, you please! The only difference is that range returns a Python list object and xrange returns an xrange object.

This means that xrange doesn’t actually generate a static list at runtime as range does. It creates the values as you need them with a special technique called yielding. This technique is used with a type of object known as generators. That means that if you have a gigantic range, you’d like to generate a list for, say one billion, xrange is the function to use.

This is especially true if you have a really memory sensitive system such as a cell phone that you are working with, as range will use as much

memory as it can to create your array of integers, which can result in a memory error and crash your program. It's a memory hungry beast.

Is Python casesensitive?

A language is case sensitive if it distinguishes between identifiers like myname and Myname . In other words, it cares about case: lowercase or uppercase. Let's try this with Python.

```
myname='Sameer'

Myname
```

Traceback (most recent call last):

File "", line 1, in

Myname

NameError: name 'Myname' is not defined

As you can see, this raised a NameError . This means that Python is indeed case sensitive.

How would you declare a comment in Python?

Unlike languages like C++, Python does not have multi line comments. All it has is octothorpe (#). Anything following a hash is considered a comment, and the interpreter ignores it.

```
# line 1 of comment

# line 2 of comment
```

In fact, you can place a comment anywhere in your code. You can use it to explain your code.

Mention five benefits of Python.

Python comprises a huge standard library for most internet platforms like email, HTML, and so on.

Python does not require explicit memory management as the interpreter itself allocates the memory to new variables and free them automatically.

Python provides easy readability because of square brackets.

Easy-to-learn for beginners.

Having the built-in data types saves programming time and effort from declaring variables.

Data types and data structures

What data types does Python support?

This is the most basic Python interview question. Python provides us with five kinds of data types:

Numbers: Numbers are used to hold numerical values. For example:

```
a=7.0
```

Strings: A string is a sequence of characters. We declare it using single or double quotes. For example:

```
title="Sameer's Book"
```

Lists: A list is an ordered collection of values, and we declare it using square brackets. For example:

```
colors=['red','green','blue']

type(colors)
```

Tuples: A tuple, which is like a list, is an ordered collection of values. The difference, however, is that a tuple is immutable. This means that we

cannot change a value in it. For example:

```
name=('Sameer','Paradkar')

name[0]='Avery'
```

Traceback (most recent call last):

File “”, line 1, in

name[0]=’Avery’

TypeError: ‘tuple’ object does not support item assignment

Dictionary: A dictionary is a data structure that holds key-value pairs. We declare it using curly braces. For example:

```
squares={1:1,2:4,3:9,4:16,5:25}

type(squares)

type({})
```

We can also use a dictionary comprehension:

```
squares={x:x**2 for x in range(1,6)}

squares
```

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

How to get indices of N maximum values in a NumPy array?

We can get the indices of N maximum values in a NumPy array using the following code:

```
import numpy as np

arr = np.array([1, 3, 2, 4, 5])

print(arr.argsort()[-3:][::-1])
```

The output for the preceding code is [4 3 1] .

How do you calculate percentiles with Python/NumPy?

We can calculate percentiles with the following code:

```
import numpy as np

a = np.array([1,2,3,4,5])

p = np.percentile(a, 50)  #Returns 50th percentile, e.g. median

print(p)
```

The output for the previous code is: 3

What advantages do NumPy arrays offer over (nested) Python lists?

Python’s lists are efficient general-purpose containers. They support (fairly) efficient insertion, deletion, appending, and concatenation, and Python’s list comprehensions make them easy to construct and manipulate.

They have certain limitations. They don’t support vectorized operations like element-wise addition and multiplication, and the fact that they can contain objects of differing types mean that Python needs to store type information for every element and need to execute type dispatching code when operating on each element.

NumPy is not just more efficient; it is also more convenient. You get many vector and matrix operations for free, which sometimes allow you to avoid unnecessary work. They are also efficiently implemented.

NumPy array is faster and you get a lot built in capabilities with NumPy , FFTs, convolutions, fast searching, basic statistics, linear algebra, histograms, and so on.

Explain the use of decorators.

Decorators in Python are used to modify or inject code in functions or classes. Using decorators, you can wrap a class or function method call so that a piece of code can be executed before or after the execution of the original code. Decorators can be used to check for permissions, modifying or tracking the arguments passed to a method, logging the calls to a specific method, and so on.

What is the difference between NumPy and SciPy?

In an ideal world, NumPy would contain nothing but the array data type and the most basic operations such as indexing, sorting, reshaping, basic element-wise functions, and so on.

All numerical code would reside in SciPy . However, one of NumPy's important goals is compatibility, so NumPy tries to retain all features supported by either of its predecessors.

Thus, NumPy contains some linear algebra functions, even though these belong to SciPy. In any case, SciPy contains more fully featured versions of the linear algebra modules, as well as many other numerical algorithms.

If you are doing scientific computing with Python, you should probably install both NumPy and SciPy . The new features are provided in to SciPy rather than NumPy .

How do you make 3D plots/visualizations using NumPy/SciPy?

Like 2D plotting, 3D graphics is beyond the scope of NumPy and SciPy, but in the 2D case, packages exist that integrate with NumPy. Matplotlib provides basic 3D plotting in them plot3d subpackage, whereas Mayavi provides a wide range of high-quality 3D visualization features, utilizing the powerful VTK engine.

What is pickling and unpickling?

The Pickle module accepts any Python object, converts it into a string representation and dumps it into a file using the dump function; this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

Mention the rules for local and global variables in Python.

Local variables: If a variable is assigned a new value anywhere within the function's body, it's assumed to be local.

Global variables: Those variables that are only referenced inside a function are implicitly global.

How will you share global variables across modules?

To do this for modules within a single program, we create a special module, and then import the config module in all modules of our application. This allows the module to be global to all modules.

Functions

What is a function?

When we want to execute a sequence of statements, we can give it a name. Let's define a function to take two numbers and return the greater number.

```
def greater(a,b):
```

return a is a>b else b

```
greater(3,3.5)
```

3.5

You can create your own function or use one of Python's many built-in functions.

Explain lambda expressions. When would you use one?

When we want a function with a single expression, we can define it anonymously. A lambda expression may take an input and return a value. To define the preceding function as a lambda expression, we need to type the following code in the interpreter:

```
(lambda a,b:a if a>b else b)(3,3.5)
```

3.5

Here, a and b are the inputs. a if a>b else b is the expression to return. The arguments are 3 and 3.5 . It is possible to not have any inputs here.

```
(lambda :print("Hi"))()
```

Hi

What is recursion?

When a function makes a call to itself, it is termed recursion. In order to avoid forming an infinite loop, we must have a base condition. Let's take a look at an example:

```
def facto(n):
```

if n==1: return 1

return n*facto(n-1)

```
facto(4)
```

24

Explain Python's parameterpassing mechanism.

To pass its parameters to a function, Python uses pass-by-reference. If you change a parameter within a function, the change is reflected in the calling function. This is its default behavior. However, when we pass literal arguments like strings, numbers, or tuples, they pass by value. This is because they are immutable.

How do you create your own package in Python?

We know that a package may contain subpackages and modules. A module is nothing but Python code. To create a package of our own, we create adirectoryand create a **init**.pyfile in it. We leave it empty. Then, in that package, we create amodule(s) with whatever code we want. For a detailed explanation with pictures, you can refer to Python packages.

How would you convert a string into an int in Python?

If astringcontains only numerical characters, you can convert it into an integer using the int() function:

```
int('227')
```

227

Let's check the types:

```
type('227')
```

```
type(int('227'))
```

How do you take an input in Python?

For taking an input from the user, we have the input() function. In Python 2, we had another raw_input() function.

The input() function takes, as an argument, the text to be displayed for the task:

```
a=input('Enter a number')
```

Enter a number7

But if you have paid attention, you know that it takes the input in the form of a string:

```
type(a)
```

Multiplying this by 2 gives us this:

```
a*=2
a
```

'77'

So, what if we need to work on an integer instead? We use the int() function for this.

```
a=int(input('Enter a number'))
```

Enter a number7

Now, when we multiply it by 2, we get this:

```
a*=2
a
```

14

How would you generate a random number in Python?

To generate a random number, we import the random() function from the random module.

```
from random import random
random()
```

0.7931961644126482

Let's call for help on this.

```
help(random)
```

random(…) method of random.Random instance

random() -> x in the interval [0, 1).

This means that it will return a random number equal to or greater than 0 and less than 1.

We can also use the randint() function. It takes two arguments to indicate a range from which to return a random integer:

```
from random import randint
randint(2,7)
```

6

```
randint(2,7)
```

5

```
randint(2,7)
```

7

```
randint(2,7)
```

6

```
randint(2,7)
```

2

How will you capitalize the first letter of a string?

You can capitalize the first letter of a string using the capitalize() method. Let's take a look at the following example:

```
'sameer'.capitalize()
```

‘Sameer’

```
type(str.capitalize)
```

However, it will allow other characters be unchanged, let's try this:

```
'@yushi'.capitalize()
```

‘@yushi’

How will you check whether all characters in a string are alphanumeric?

For this, we use the isalnum() method. Let’s take a look at the following example:

```
'Sameer123'.isalnum()
```

True

```
'Sameer123!'.isalnum()
```

False

The other methods that we have included are as follows:

```
'123.3'.isdigit()
```

False

```
'123'.isnumeric()
```

True

```
'sameer'.islower()
```

True

```
'Sameer'.isupper()
```

False

```
'Sameer'.istitle()
```

True

```
' '.isspace()
```

True

```
'123F'.isdecimal()
```

False

What is the usage of the help() and dir() functions in Python?

The help() and dir() functions are accessible from the Python interpreter and used for viewing a consolidated dump of built-in functions.

help() : The help() function is used to display the documentation string and also facilitates you to see the help related to modules, keywords, attributes, and so on.

dir() : The dir() function is used to display the defined symbols.

What is a docstring?

A docstring is a documentation string that we use to explain what a construct does. We place it as the first thing under a function, class, or a method to describe what it does. We declare a docstring using three sets of single or double quotes.

```
def sayhi():
```

"""

The function prints Hi

"""

print("Hi")

```
sayhi()
```

Hi

To get a function's docstring, we use its **doc** attribute.

```
sayhi.doc
```

'\n\tThis function prints Hi\n\t'

A docstring, unlike a comment, is retained at runtime.

What is the map function in Python?

The map function executes the function given as the first argument on all the elements of the iterable given as the second argument. If the function given takes in more than one argument, then many iterables are given.

What is TkInter?

TkInter is Python library. It is a toolkit for the GUI development. It provides support for various GUI tools or widgets (such as buttons, labels, text boxes, radio buttons, and so on) that are used in GUI applications. The common attributes include dimensions, colors, fonts, cursors, and so on.

Miscellaneous topics

What are the tools that help to find bugs or perform static analysis?

PyChecker is a static analysis tool that detects the bugs in the Python source code and warns about the style and complexity of the bug. Pylint is another tool that verifies if the module meets the coding standard.

What is namespace in Python?

In Python, every name introduced has a place where it lives and can be looked-up. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched, this box will be searched to get the corresponding object.

What is lambda in Python?

It is a single expression anonymous function often used as an inline function.

Why lambda forms in Python do not have statements?

A lambda form in Python does not have statements as it is used to make new function objects and then returns them at runtime.

What is pass in Python?

Pass means, no operation Python statement, or in other words it is a placeholder in the compound statement, where there should be a blank left and nothing has to be written there.

What is slicing in Python?

A mechanism to select a range of items from sequence types like lists, tuples, strings, and so on is known as slicing.

What are generators in Python?

The way of implementing iterators are known as generators. It is a normal function except that it yields expressions in the function.

What is a Python module?

A module is a Python script that generally contains import statements, functions, classes and variable definitions, and Python runnable code and it lives file with a .pyextension . ZIP files and DLL files can also be modules. Inside the module, you can refer to the module name as a string that is

stored in the global variable name.

How do you perform pattern matching in Python?

Regular expressions/REs/ regexes enable us to specify expressions that can match specific parts of a given string. For instance, we can define a regular expression to match a single character or a digit, a telephone number, or an email address, etc. The Python's re module provides regular expression patterns and it was introduced from later versions of Python 2.5. The re module provides methods for search text strings or replaces text strings along with methods for splitting text strings based on the pattern defined.

Is Python object-oriented? What is object-oriented programming?

Yes. Python is OOPs language. OOP is the programming paradigm based on classes and instances of those classes called objects . The features of OOP are encapsulation, data abstraction, inheritance, and polymorphism.

Name a few methods that are used to implement Functionally-oriented Programming in Python.

Python supports methods (called iterators in Python3) such as filter(), map(), and reduce() that are very useful when you need to iterate over the items in a list, create a dictionary, or extract a subset of a list.

filter(): It enables you to extract a subset of values based on conditional logic.

map(): It is a built-in function that applies the function to each item in an iterable (node).

reduce(): It repeatedly performs a pair-wise reduction on a sequence until a single value is computed.

How to save an image locally using Python whose URL address I already know?

We will use the following code to save an image locally from an URL address:

```
import urllib.request

urllib.request.urlretrieve("URL", "local-filename.jpg")
```

How can you get the Google cache age of any URL or web page?

Use the following URL format:

http://webcache.googleusercontent.com/search?q=cache:URLGOESHERE

Be sure to replace URLGOESHERE with the proper web address of the page or site whose cache you want to retrieve and see the time. For example, to check the Google Webcache age of edureka.co, you can use the following URL:

http://webcache.googleusercontent.com/search?q=cache:edureka.co

You are required to scrap data from IMDb top 250 movies page. It should only have fields such as movie name, year, and rating.

We will use the following lines of code:

```
from bs4 import BeautifulSoup

import requests

import sys

url = ' http://www.imdb.com/chart/top '

response = requests.get(url)

soup = BeautifulSoup(response.text)

tr = soup.findChildren("tr")

tr = iter(tr)

next(tr)

for movie in tr:
```

```
title = movie.find('td', {'class': 'titleColumn'} ).find('a').contents[0]

year = movie.find('td', {'class': 'titleColumn'} ).find('span', {'class': 'secondaryInfo'}).contents[0]

rating = movie.find('td', {'class': 'ratingColumn imdbRating'} ).find('strong').contents[0]

row = title + ' - ' + year + ' ' + ' ' + rating

print(row)
```

The preceding code will help scrap data from IMDb's top 250 list.

Mention the differences between Django, Pyramid, and Flask.

Flask is a microframework primarily build for a small application with simpler requirements. In flask, you have to use external libraries. Flask is ready to use.

Pyramid is built for larger applications. It provides flexibility and allows the developer to use the right tools for their project. The developer can choose the database, URL structure, template style, and more. Pyramid is heavy configurable.

Django can be used for larger applications just like Pyramid. It includes an ORM.

Discussthe Django architecture.

The following diagram explains the Django MVT pattern:

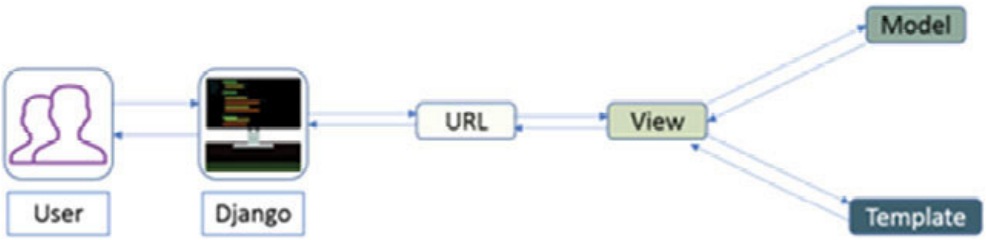


Figure 10.1: Python – Django architecture

The developer provides the model, the view and the template and then just maps it to a URL and Django does the magic to serve it to the user.

Explain how you can set up the database in Django.

You can use the editmysite/setting.py command. It is a normal Python module with a module level representing Django settings.

Django uses SQLite by default; it is easy for Django users, as it won't require any other type of installation. In case your database choice is different that you have to the following keys in the DATABASE defaultitem to match your database connection settings:

Engines: You can change the database using django.db.backends.sqlite3, django.db.backeneds.mysql, django.db.backends.postgresql_psycopg2, django.db.backends.oracle, and so on

Name: This is the name of your database. If you are using SQLite as your database, in that case the database will be a file on your computer, the name should be a full absolute path, including file name of that file.

If you are not choosing SQLite as your database, then the settings like password, host, user, and so on must be added.

Django uses SQLite as the default database; it stores data as a single file in the filesystem. If you do have a database server—PostgreSQL, MySQL, Oracle, MSSQL—and you want to use it rather than SQLite, use your database's administration tools to create a new database for your Django project. Either way, with your (empty) database in place, all that remains is to tell Django how to use it. This is where your project's settings.py file comes in.

We will add the following lines of code to the .py file setting:

```
DATABASES = {
    'default': {
        'ENGINE' : 'django.db.backends.sqlite3',
```

```
        'NAME' : os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

Give an example of how you can write a VIEW in Django.

This is how we can write a view in Django:

```
from django.http import HttpResponse
import datetime
def Current_datetime(request):
    now = datetime.datetime.now()
    html = "It is now %s" % now
    return HttpResponse(html)
```

The preceding code returns the current date and time, as an HTML document.

What does the Django templates consist of?

The template is a simple text file. It can create any text-based format like XML, CSV, HTML, and so on. A template contains variables that are replaced with values when the template is evaluated and tags (% tag %) that control the logic of the template. The following diagram depicts the Python Django template:

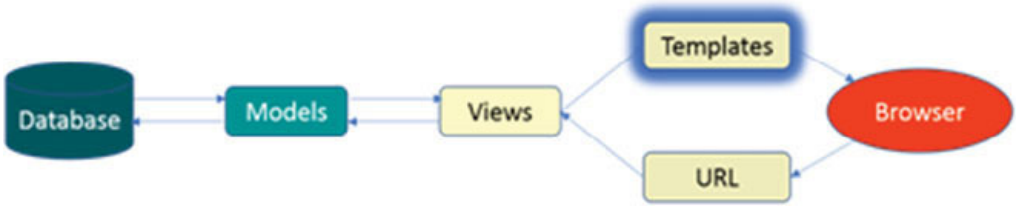


Figure 10.2: Python – Django template

Explain the use of a session in the Django framework.

Django provides a session that allows you to store and retrieve data on a per-site-visitor basis. Django abstracts the process of sending and receiving cookies by placing a session ID cookie on the client side, and storing all the related data on the server side. The following diagram depicts the Django framework:

Figure 10.3: Python – Django framework

Therefore, the data itself is not stored on the client side. This is nice from a security perspective.

List the inheritance styles in Django.

In Django, there are three possible inheritance styles:

Abstract base classes: This style is used when you only wantthe parent's class to hold information that you don't want to type out for each child model.

Multi-table inheritance: This style is used if you are subclassing an existing model and each model needs to have its own database table.

Proxy models: You can use this model if you only want to modify the Python level behavior of the model, without changing the model's fields.

How do you debug a program in Python?

To debug a Python program, we use the pdb module. This is the Python debugger; we will discuss it in a tutorial soon. If we start a program using pdb , it will allow us to step through the code.

Explain garbage collection with Python.

The following points are worth nothing for the garbage collector with Python:

Python maintains a count of how many references there are to each object in memory.

When a reference count drops to zero, it means that the object is dead and Python can free the memory it allocates to that object.

The garbage collector looks for reference cycles and cleans them up.

Python usesheuristicsto speed up garbage collection.

Recently created objects might as well be dead.

The garbage collector assigns generations to each object as it is created.

It deals with the younger generations first.

How is Python different from Java?

The following is thecomparison of Python vs Java:

Java is faster than Python.

Python mandates indentation. Java needs braces.

Python is dynamically-typed; Java is statically typed.

Python is simple and concise; Java is verbose

Python is interpreted.

Java is platform-independent.

Java has a stronger database-access with JDBC.

How do you open a file for writing?

Let's create a text file on our desktop and call it tabs.txt . To open it and to be able to write to it, use the following line of code:

```
file=open('tabs.txt','w')
```

This opens the file in the writing mode. You should close it once you're done.

What is a control flow statement?

A Python program usually starts to execute from the first line. From there, it moves through each statement just once and as soon as it's done with the last statement, it terminates the program. However, sometimes, we may want to take a more twisted path through the code. Control flow statements allow us to disturb the normal execution flow of a program and bend it to our will by adding control flow constructs for addressing the real-world problems.

R programming

R is one of the most popular programming languages for Big Data professionals. There is a huge requirement of R programmers in the market and this section covers the concepts and usually asked interview questions to R programmers by the interviewers. R programming language is used for data analysis and developing statistical software. R programming language is also used for machine learning purpose and applications.

Core concepts

What is R programming?

R is a programming language and a software environment meant for statistical analysis and creating graphs. It is used by analysts, statisticians and data scientists for various purposes. R uses data objects for data calculations and it is an alternative to conventional statistical packages such as SAS, SPSS, and so on. Many companies are incorporating R into their business models to proliferate their revenue. There aremany career prospects in R such as data scientist, R programmer, analyst consultant, and many more.

What are the features of R programming?

The following are the programming features of R:

There are packages in R. These R packages are useful in collecting functions into a single set.

R programming includes database input, data export, variable labels, and so on.

R is an interpreted language and support matrices.

R supports object-oriented programming and procedural programming. Object-oriented programming consists of classes and objects, whereas procedural programming includes records and procedures.

What are the applications of R?

The scope of R as a programming language is high and it has varied applications in various verticals. The important applications are as follows:

R is used as an important tool in finance. It is used by several data analysts and research programmers.

R deals with a lot of statistics. It is considered the best for data science. R also provides an environment for statistical computing and designing.

R is also used for data importing and cleaning.

Many data analysts and research programmers use R because R is the most prevalent language. Hence, R is used as a fundamental tool for finance.

Many quantitative analysts use R as their programming tool. Hence, R helps in data importing and cleaning, depending on the strategy you are using.

R is best for data science because it gives a broad variety of statistics. In addition, R provides the environment for statistical computing and designing. R is considered an alternate execution of S.

What is a GUI in R?

GUI stands for Graphical User Interface. R is a command line driven program. The user enters commands at the prompt (> by default) and each command is executed one at a time. There have been a number of attempts to create more graphical interfaces, ranging from code editors that interact with R to full-blown GUIs that present the user with menus and dialog boxes.

What is a CLI in R?

CLI stands for Command Line Interface. In a command line interface, you type commands that you want to execute and press return. For example, if you type the line 2+2 and press the return key, R will give you the result

Why should you adopt R programming language?

R programming language is best for statistical, data analysis and machine learning. Using this language, we can create objects, functions, and packages. R is an open source programming language.

Using R, we can create any form of statistics and data manipulation. Further more, it can be used in almost every field of finance, marketing, sports, and so on. R Programming is extensible and hence, R groups are noted for its energetic contributions.

Many typical features can be written in R itself and hence, R has gotten faster over time and serves as a glue language.

What are programming features of R?

The following are the features of R programming:

Packages are part of R programming. Hence, they are useful in collecting sets of R functions into a single unit.

R’s programming features include database input, exporting data, viewing data, variable labels, missing data, and so on.

R is an interpreted language. So, we can access it through the command line interpreter.

R supports matrix arithmetic.

R supports procedural programming with functions and object-oriented programming with generic functions. Procedural programming includes procedure, records, modules, and procedure calls while object-oriented programming language includes class, objects, and functions.

Compare R with other technologies.

The following points compare R with other technologies:

Data handling capabilities: R has good data handling capabilities and options for parallel computation.

Availability/cost: R is an open source and we can use it anywhere.

Advancement in tool: If you are working on the latest technologies, R gets the latest features.

Ease of learning: R has a learning curve. R is a low-level programming language. As a result, simple procedures can take long codes.

Job scenario: It is a better option for startups and companies looking for cost efficiency.

Graphical capabilities: R has the most advanced graphical capabilities. Hence, it provides us advanced graphical capabilities.

Customer service support and community: R is the biggest online growing community.

Why is R important?

R is a leading tool formachine learning, statistics, and data analysis. It is a programming language. Using R, we can create objects, functions, and packages. R language is platform independent, so we can use it on any operating system. The installation of R is free, so we can use it without purchasing a license. R is not only a statistical package but also an open source language. It means anyone can examine the source code to see what exactly is doing on the screen. Anyone can add a feature and fix bugs without waiting for the vendor to do this. Thus, it allows you to integrate with other languages (C, C++). It also enables you to interact with many data sources and statistical packages (SAS, SPSS). R has a large growing community of users.

What are the advantages and disadvantages of R?

The advantages of R programming are as follows:

R is the most comprehensive statistical analysis package as new technologies and ideas often appear first in R.

R is an opensource software. Hence, anyone can use andchange it.

R is an open source. We can run R anywhere and at any time, and even sell it under conditions of the license.

R is good for GNU/Linux and Microsoft Windows. R is cross-platform, which runs on many operating systems.

In R, anyone is welcome to provide bug fixes, code enhancements, and new packages.

The disadvantages of R programming are as follows:

In R, the quality of some packages is less than perfect.

In R, no one to complain, if something doesn’t work.

R is a softwareapplication that many people devote their own time to developing.

R commands give little thought to memory management, and so R can consume all the available memory.

What is OOP in R?

Object-oriented programming is a popular programming language. It allows you to construct modular pieces of code, which are used as building blocks for large systems. R is a functional language. It also supports programming in an object-oriented style. OOP is a superb tool to manage complexity in larger programs. It is particularly suited to GUI development. R has two different OOP systems known as S3 and S4.

What is procedural programming in R?

Procedural programming is a programming paradigm, derived from structured programming, based on the concept of the procedure call.

Procedures, also known as routines, subroutines, or functions (not to be confused with mathematical functions, but similar to those used in functional programming), simply contain a series of computational steps to be carried out. Any given procedure might be called at any point during a program's execution, including other procedures or itself.

Data types and data structures

What are the different data structures in R?

Data structure is a form of organizing and storing data. It is imperative to have a strong understanding of various data types and data structures in order to make the best use of R language. R programming supports five types of data structures such as vector, matrix, list, data frame, and factor:

Vector: This data structure contains an integer, double, complex, and so on. It is a sequence of the same data elements and the c() function is used to create a vector in R programming.

Matrix: It is a two-dimensional data structure and is used to bind vectors from the same length. All the elements in the matrix have to be of the same type and it is created using a matrix() function. The value of the row can be defined using nrow and the value of the column can be defined using ncol.

List: It includes data of different types like numbers, strings, vectors, and so on. It is somewhat like a vector, but it contains mixed elements. A list is created using () .

Data frame: It is a special list where each element is of the same length. A data frame has the features of both, matrices and lists. It is more generic than the matrix as different columns have different data types. It is created using the frame() function.

Factors: It is created using the factor() function and is used to store predefined and categorical data.

How can you import data in R?

There are several ways to import data in R. You can use the R commander to import data in R:

Excel file: If the sample data is in the Excel format, the read.xls function is used from the data package. It returns a data frame. Alternatively, loadWorkbook can also be used to read the entire workbook.

Minitab file: If the data file is in the Minitab format, it can be opened using the read.mtb function. It returns a list of components in the Minitab worksheet.

SPSS file: The data files in SPSS formats can be opened using the read.spss function. It returns a list of components.

CSV file: Each cell inside the CSV data file is separated by a special character such as a comma.

How can we convey the data analysis result through R language?

The user can combine the result of data, code, and analysis in a single document and the processed data can be used by the users to perform reproducible research. In this way, the user can verify the result and engage in many discussions. Through such reproducible research, the experiments can be done easily and applied to different problems.

What are R data types? How many types of data types are provided by R?

In programming, a data type is a classification that specifies what type of a value a variable has. It also describes what type of relational, mathematical and logical operations can apply to it without causing an error. We need to use various variables to store information while doing programming in any programming language. Variables are nothing but reserved memory locations to store values. This means that when we create a variable, we reserve some space in memory. The variables are assigned with R objects. Thus, the data type of the R object becomes the data type of the variable.

There are five types of data types present in R:

Integer data type

Numeric data type

Character data type

Complex data type

Logical data type

List the data structures used in R programming.

There are two data structures in R programming: one is Homogeneous and other is Heterogeneous. When the same type of objects are to be used, then homogeneous data structures are used as for vectors, arrays and matrix. Even for different object types, we can use heterogeneous data structures to store data frames and lists.

Explain different types of objects provided by R.

The different types of objects provided by R programming are as follows:

Vectors: A vector is a sequence of data elements of the same basic type. Hence, members of a vector are called components. Here is a vector containing three numeric values 2, 3, and 5.

Matrices: It is a collection of numbers. This is arranged into a fixed number of rows and columns. Generally, the numbers are real numbers.

Array: While matrices are confined to two dimensions, arrays can be of any number of dimensions. Arrays are the R data objects. We use these data objects to store data in more than two dimensions.

Lists: A list is a generic vector containing other objects.

Data frames: Data frames are tabular data objects. Unlike the matrix in a data frame, each column can contain different modes of data.

Factors: In R, we store factors as a vector of integer values with a corresponding set of character values to use when the factor is displayed. Moreover, we use the factor function to create a factor. Therefore, the only required argument to factor is a vector of values, which will return as a vector of factor values.

What is data structure in R?

A data structure is a specialized format for organizing and storing data. General data structure types include the array, the file, the record, the table, the tree, and so on. Any data structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways

Functions and packages

What is function definition?

An R function is been created using the keyword function . The basic syntax of an R function definition is as follows:

function*name* <- *function(arg*1, arg_2, …) {

  Function body

}

What are the components of R functions?

The different parts of a function are as follows:

Function name: It is the actual name of the function because it is stored in the R environment as an object with this name.

Arguments: An argument is a placeholder. When a function is invoked, we pass a value to the argument. Arguments are optional; that is, a function may contain no arguments. Also, arguments can have default values.

Functions body: In a function body, statements can be collected. And hence, it defines what the function does.

Return value: The return value of a function is the last expression in the function body to check.

Enlist some of the functions that R provides.

The functions that R provides are as follows:

Mean: It is calculated by taking the sum of the values and dividing it by a number of values. The function used is mean() .

Median: It is the middle most value in the data series. The function used in R programming is median() .

The other functions of R include Regression, GLM, mixed-effects, distribution, GAM, non-linear, and so on.

How many sorting algorithms are available?

In R, the following sorting algorithms are available:

Bubble sort

Selection sort

Merge sort

Quick sort

Bucket sort

What is the value of f(2) for the following code?

```
a<-4

f <- function (b)

{

 a<- 3

 a^3 + g (b)

}

g<-function (b)

{a*b}
```

In the preceding function, the value 2 will be passed to the function, which is for variable b and the value of variable a will be defined in the f(b) function that is 3 . So, the output of the preceding program will be 3^3+g(2) . In the preceding code, the value of function g is global and it takes value 4 rather than 3 and will return the value 2*4 -8 to the function f. Here, the effective result will be 3^3+8=35 .

What is data import in R language?

The user can import data in R language. The R-commander GUI is used to type the commands, also known as Rcmdr , which is like a console.

What are R functions?

A function is a piece of code written to carry out a specified task. Thus, it can or can't accept arguments or parameters and it can or can't return one or more values. In R, functions are objects in their own right. Hence, we can work with them exactly the same way we work with any other type of object.

What is the attribute function in R?

To get or set a single attribute, you can use the attr() function. This function takes two important arguments. The first argument is the object we want to examine, and the second argument is the name of the attribute we want to see or change. If the attribute we ask for doesn't exist, R simply returns NULL.

What is the length function in R?

The length() function gets or sets the length of a vector (list) or other objects. The length() function can be used for all R objects. For an environment, it returns the object number in it. NULL returns 0.

What are generic functions in R?

R has three object-oriented (OO) systems: [[S3]], [[S4]] and [[R5]]. A method is a function associated with a particular type of object. S3 implements a style of object-oriented programming called generic-function OO.

What are R packages?

Packages are collections ofR functions, data, and compiled code in a well-defined format. The directory where packages are stored is called the

library. R comes with a standard set of packages. The others are available for download and installation. Once installed, they have to be loaded into the session to be used.

What is the recursive function in R?

Recursive functions call themselves. They break down the problem into the smallest possible components. The function() calls itself within the original function() on each of the smaller components. After this, the results will be put together to solve the original problem. For example:

```
Factorial <- function(N)

{

 if (N == 0)

   return(1)

 else

   return( N * Factorial (N-1))

}
```

What are packages in R?

R packages are a collection of R functions, complied code and sample data. They are stored in a directory called library in the R environment. By default, R installs a set of packages during installation.

What are the different packages in R?

R packages are a collection of R functions and sample data is stored under a directory name called library. Initially, R adds a set of packages during installation. However, new packages are added as and when required for specific purposes. The different packages available in R are as follows:

MICE: The MICE package deals with missing data. It creates replacement values for the missing data. There are two types of missing data: MCAR and MNAR. In this package, the mice() function looks after the inputting process.

Amelia: This package is used for multiple imputations of missing data. It also produces multiple output datasets for analysis. To use this package, you can either invoke the ameliagui() command or run the Amelia function on the data.

Mi: The Mi package provides functions for data manipulation and imputes missing values. This package has several features that allow the users to get into the imputation process and gauge the reasonableness of the resulting model.

What packages are used for data mining in R?

The packages used for mining in R are as follows:

Data.table: It supports fast reading of large files.

Arules: It is used for rule learning.

Tm: It is used to perform text mining.

Forecast: It provides functions for time series analysis.

Miscellaneous topics

Why is clustering required in data analysis?

Clustering refers to a group of objects that belong to the same class. It is a process to make a group of abstract objects in the class of similar objects. Clustering is required in data analysis due to the following reasons:

Scalability: Clustering is required to deal with large databases.

Interpretability: The result of clustering should be comprehensive and usable.

Dimensionality: The clustering algorithm is used to handle high-dimensional space.

Deal with noisy data: Databases contains erroneous data. Algorithms that are sensitive to such data may deliver poor results.

What is a white noise model in R?

In R, a white noise model is a basic time series model, which is also the basis for more elaborated and defined models. To stimulate the data from a variety of tie series model, the Arima.sim() function is used. The white noise model has a fixed constant mean, fixed constant variance and no correlation over time.

What is the random walk model in R?

In R programming, the random walk model is an example of the non-stationary model. A random walk has no fixed mean or variance. It also has a strong dependence over time. There are two types of random walks, namely, random walk without drift and random walk with drift.

Compare R and Python.

The following are a few differences between Python and R language:

What is visualization in R?

Visualization is any technique for creating images, diagrams, or animations to communicate a message. Visualization through visual imagery has been an effective way to communicate both abstract and concrete ideas since the dawn of humanity.

Conclusion

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently, whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is a high level object-oriented programming language. There are many benefits of using Python. Firstly, Python scripts are simple, shorter, portable and open-source. Secondly, Python variables are dynamic typed. So, you don't need to think about the variable type while coding. Thirdly, Python classes have no access modifiers, which Java has. So, you don't need to think about access modifiers. Lastly, Python provides us different libraries and datastructures to make our coding easier.

R is an open source programming language and software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. R is a programming language and free software environment used for multiple purposes such as statistical analysis, data manipulation, predicting and forecasting, etc. With R, well-designed publication plots can be produced. R runs on platforms like UNIX, LINX, Windows, and macOS. The code for R is written in C, Fortran and R. R is an interpreted language that can implement a wide variety of statistical and graphical techniques. R makes it easier for the users to follow the algorithm choices as most of the functions are written in R itself.

Due to it interesting benefits, R is used by several companies such as Google, Facebook, Ford, and so on. R is used by the Human Rights Data Analysis Group to gauge the impact of war. R is also used by Ford to revamp the designs of its vehicles. R has a promising future because of its open source nature. According to Gartner, the popularity of R will definitely grow further. So, it is the right time to move forward in your career with R.

This chapter covered Q & A for trending programming languages, which include Python and R. This chapter presented an exhaustive question bank with special emphasis on practical scenarios. This chapter covered topics such as programming constructs, datatypes and variables, OOPS, data structures, functions, and procedures. The next chapter will present an exhaustive question bank that will cover aspects related to validation and testing, DevOps, Agile, Scrum, SDLC, microservices, and SOA.

CHAPTER 11

Methodologies and Processes

In software engineering, a software development methodology also known as software development lifecycle is segregating of software development into distinct phases consisting of activities with the intent of better planning and management. It is often considered a subset of the system development lifecycle. The methodology includes the pre-definitions of specific deliverables and artifacts that are created and completed by a team to develop or maintain a software application.

Common methodologies include a waterfall, iterative and incremental development, spiral development, rapid application development, extreme programming and Agile and scrum methodology. A few consider that the lifecycle model is a term for a category of methodologies and a software development process a more specific general term to refer to a process selected by a specific company. For example, there are many specific software development processes that fit the spiral lifecycle model. This chapter presents an exhaustive question bank that covers aspects pertaining to SDLC, verification and validation activities, different methodologies, DevOps, Agile/Scrum, and configuration management.

Software development methodologies – Agile and Scrum

What are different types of software development methodologies?

Some of the commonly used software development methodologies are:

The Waterfall model

Iterative development

The V model

The Agile model

The Spiral model

What is a waterfall model?

The waterfall model describes a development method that is linear and sequential. The waterfall development model has distinct goals for each phase of development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back.

What are advantages of using a waterfall model?

The advantages of a waterfall model are that it allows departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process. Development moves from requirement, through design, implementation, testing, installation, troubleshooting, and ends up with operation and maintenance. Each phase of development proceeds in a strict order, without any overlapping or iterative steps.

What are the disadvantages of using a waterfall model?

The disadvantage of the waterfall development model is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well thought of in the requirement stage.

What is an iterative model?

An iterative lifecycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model.

What is Agile?

Agile is a group of software development methods based on iterative and incremental development where requirements and solutions evolve through collaboration between self-organizing and cross-functional teams.

Agile processes and methods promote a disciplined project management process that emphasizes frequent inspection and adaptation, a leadership philosophy that encourages self-organization, teamwork and accountability, engineering best practices intended to allow rapid delivery of high-quality software, and a business approach that aligns development with customer requirements and organization goals. Agile development refers to a development process that is aligned with the concepts of Agile Manifesto. The following diagram depicts the Agile methodology:

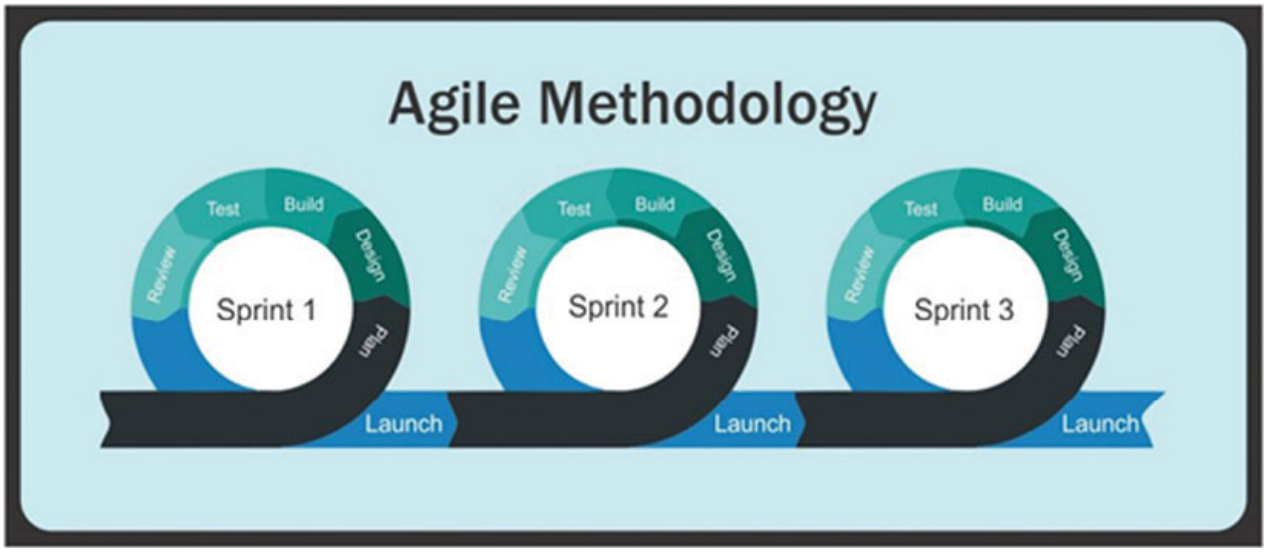


Figure 11.1: Agile methodology

What is a Scrum?

Scrum is a subset of Agile and is a lightweight process framework for Agile development and most widely leveraged in the industry.

A process framework is a set of practices that need to be followed for a process to be consistent with the framework. The Scrum process framework requires leveraging development cycles called Sprints , whereas the XP framework requires pair programming, and so forth.

Lightweight means that the overhead of the process is kept as small as possible to maximize the amount of productive time available for getting the useful work done.

A Scrum process is distinguished from the other Agile processes by the concepts and practices, which are divided into the three categories of roles, artifacts, and time boxes. These and other terms leveraged in Scrum are defined below. Scrum is most often leveraged to manage complex software and product development, using iterative and incremental development practices. Scrum significantly boosts the productivity and reduces time to benefit relative to classic waterfall processes. Scrum processes enable companies to adjust smoothly to rapidly changing requirements and produces software that meets evolving business demands.

The following diagram depicts the Scrum methodology:

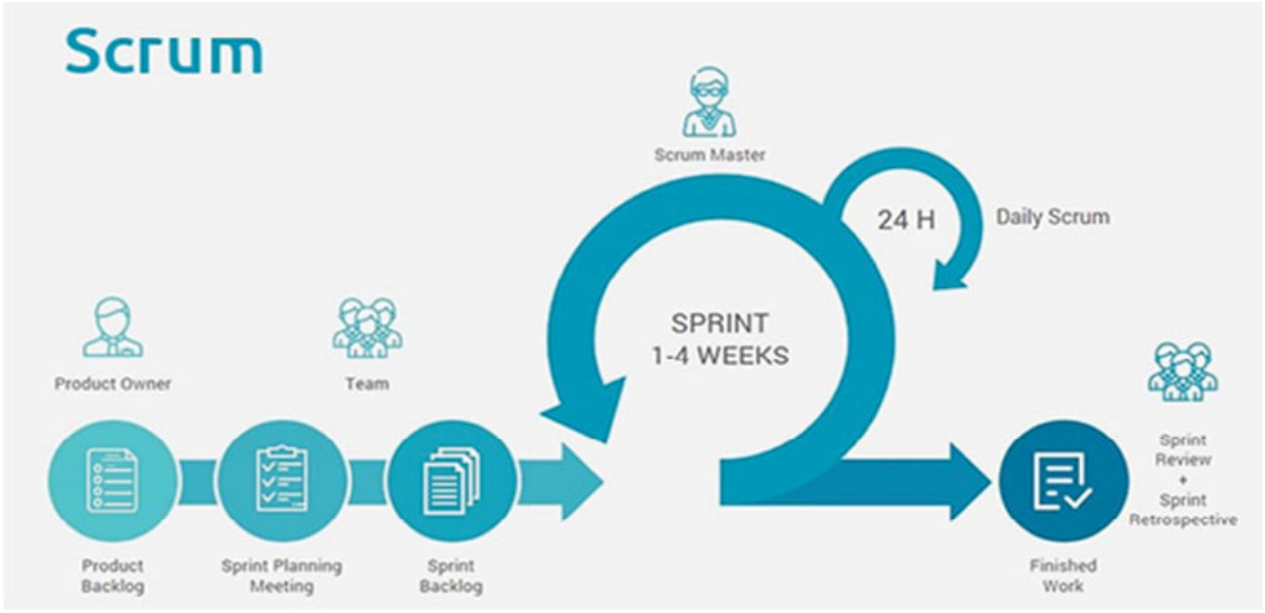


Figure 11.2: Scrum methodology

Describe benefits of using the Agile scrum process.

An Agile scrum process benefits the organization by helping it to:

Ensure quality of all the deliverables

Cope better with changes and accept changes

Provide correct estimates while spending less time creating them

Be more in control of the project schedule and state

How does Agile testing or development methodology differ from other development methodologies?

The testers or developers ensure that the whole process of testing or development is broken into small steps as possible, and just a small unit of code is tested or developed in each of these steps. The team of testers or developers consistently communicates the results of their work and changes the short-term strategy and the development plan on the go, basis the results of Agile testing. Agile methodology encourages flexible and rapid responses to changes, which should lead to better results.

What is a Sprint?

A Sprint is a regular, repeat work cycle in the scrum methodology during which the work is completed and made ready for reviews. Sprints are fundamental units of development in the scrum methodology. Typically, sprints are less than 30 days long. Sprints consist of the sprint planning, daily scrums, development work, sprint review, and sprint retrospective. During the Sprint:

No changes are made that would endanger the Sprint goal.

Quality goals do not decrease.

The scope can be clarified and renegotiated between the product owner and development team as more is learned.

What are the most important components of Agile?

The key features of Agile are as follows:

Daily stand-up meetings

Time boxed task boards

CRC/Class responsibilities and collaborators cards

Sprit planning meetings and Sprint development

Test-driven Development (TDD), Continuous Integration, regular code reviews, pair programming, automated builds, continuous deployment, and delivery, and so on

What do daily stand-up meetings entail?

Each day, at the same time and same place and in front of the task board, the team meets to give updates about their tasks and tickets resolved for the day. This meeting addresses SCRUM's three questions listed as follows:

What have you completed since the last meeting?

What do you plan to complete by the next meeting?

What is getting in your way?

What are different roles in Scrum?

Scrum prescribes three roles: the Product Owner, Scrum Master, and the Delivery team. These roles should, ideally, be cross-functional and not shared among other engagements.

What is the role of the ScrumMaster?

A ScrumMaster serves the team and shields them from the distractions that would prevent them from completing a sprint goal. They also remove blocks, teach the team to become self-organized and serve as a coach who teaches Agile and Scrum values and principles.

How can a storyboard be defined in Agile?

A storyboard is a visual representation of a software project's progress. There are generally four columns: To do, In Progress, Test, and Done .

Different colored post; its notes are placed in each column, indicating the progress of individual development items. A storyboard is typically leveraged in Agile development.

How much time should a person expect to spend on ScrumMaster activities?

A ScrumMaster should make this role the top priority to focus on benefits of the overall team. Their load will vary from sprint to sprint, depending on what impediments and issues the teams are dealing with. Newly formed teams typically take more ScrumMaster Time; 50%-100%, while experienced ScrumMasters with well-functioning teams might spend 50% or less time on the ScrumMaster role.

What is the difference between epic, feature, user stories, and tasks?

Epic: An epic is a large body of work that can be broken down into a number of related groups of functionalities. It is a group of related user stories.

Feature: A feature represents a distinct element of functionality, which can provide capabilities to the business.

User stories: User stories are descriptions of desired functionalities, which are told from the perspective of the user. These are the actual business requirement generally created by the business owner.

Task: To accomplish the business requirements, the development team creates tasks. Tasks are further decomposition of user stories, which are activities that are required to help a user story meet the Definition of Done (DoD) .

What is a product backlog?

A product backlog contains all functionalities required to meet the business goals as defined in the Program Canvas. It contains the master list of features and user stories desired for the final product. It is consistently refined by the Product Owner and development team.

What is a Sprint Backlog?

A Sprint Backlog is a group of user stories identified by the Agile team to be completed during the Sprint. It is a group of prioritized and estimated user stories decomposed into tasks, selected during Sprint Planning with input from the Product Owner.

What is Minimum Viable Product (MVP)?

Minimum Viable Product (MVP) is the smallest subset of features that can be implemented together to deliver value to the business and customer.

Explain velocity in Agile.

Velocity tracks the actual amount of work completed from sprint to sprint. This helps you determine your team's velocity and estimate the work your team can realistically achieve in future sprints.

Does maximum velocity mean maximum productivity?

No, in an attempt to maximize velocity, a team may, in fact, achieve the opposite. If asked to maximize velocity, a team may skip on the unit or acceptance testing, reduces customer collaboration, skips fixing bugs, and minimizes refactoring. While potentially offering short-term improvement, there will be a negative long-term impact. The goal is not to maximize velocity instead of the optimal velocity over time, which takes into account many factors, including quality of the end product.

What does a Scrum burndown chart comprise?

A burndown chart is a graphical representation of the estimated work that remains in the sprint. It shows the daily sprint or release status in a visual format that is based on user stories. It reflects work remaining on the Y-axis and time in X-axis. It can track either remaining hours or remaining story points. It is updated daily to reflect most of the up-to-date work remaining by sprint and/or release.

An ideal work line (the straight line from maximum hours down to the completion date) depicts the teams' average rate of production. When the remaining work line is above the ideal line, the team is behind schedule. A decrease in the remaining tasks can occur as the team makes progress or when the Product Owner removes items from the sprint

What is a release candidate?

A release candidate is a version or a build of software that can be released to production. Further, testing such as UAT may be performed on this version.

What are the differences between Agile and traditional project management?

Agile is a framework consisting of approaches and behaviors that encourage just-in-time production that enables customers to receive quality software quickly. Agile encourages that a little of everything, including design, development, and testing is done at the same time, as opposed to the traditional approach to projects, where one phase is completed before the next begins. Agile encourages frequent, short feedback loops and embraces changes to the requirements. In the waterfall model, feedback is usually not collected until the very end of the project and changes are discouraged.

Describe what happens in the Sprint planning meeting.

The Sprint planning meeting is scheduled to discuss the team goal in the upcoming Sprint. The event itself has two parts. The first half of Sprint planning is used to decide 'What' the Agile team will be working on by pulling items from the product backlog into their Sprint backlog. The second half of Sprint planning is utilized by the Agile team to determine How they will accomplish the work that they have in the Sprint backlog.

What is a Scrum of Scrums?

It is essential to manage cross-team dependency and hence scum of scrums is arranged. The scrum of scrum is:

A meeting for cluster of teams to discuss their work, which focuses especially on areas of overlap, dependencies, and integration during a sprint.

Used as a technique to scale scrum to large project teams.

Typically used when there are multiple scrum teams that require coordination or have dependencies to be managed in order to stay aligned overall.

Participants in the scrum of scrums meetings are individual contributors to their teams (tech lead, tester, analyst, scrum master, and so on).

Similar to a daily stand-up, the Scrum of Scrums is time boxed to 15 minutes; however, it can be extended as agreed by the participants.

Plan for 2-3 scrum of scrums per week, or as needed.

DevOps

Why DevOps?

DevOps is an abbreviation of words Development and Operations . DevOps is a software delivery approach, culture or practice that brings the development team and other IT stakeholders like business, testing, and operations teams together to achieve a common business goal. Some of the key aspects of DevOps are collaboration, integration, and communication between various stakeholders and automation of DevOps processes.

Instead of releasing big sets of features, companies are trying to see if small features can be transported to their customers through a series of release trains. This has many advantages like quick feedback from customers, better quality of software, and so on, which in turn leads to high customer satisfaction. To achieve this, companies are required to:

Increase deployment frequency

Lower failure rate of new releases

Shortened lead time between fixes

Faster mean time to recovery in the event of new release crashing

DevOps fulfills all these requirements and helps in achieving seamless software delivery. Companies like Etsy, Google, and Amazon, which have adopted DevOps to achieve levels of performance that were unthinkable of even five years ago. They are doing tens, hundreds or even thousands of code deployments per day while delivering world class stability, reliability, and security.

What is the DevOps lifecycle?

DevOps is deep integration between development and operations. Understanding DevOps is not possible without knowing the DevOps lifecycle.

Here is brief summary about the Continuous DevOps lifecycle:

Development: In this DevOps stage, the development of software takes place constantly. In this phase, the entire development process is separated into small development cycles. This benefits the DevOps team to speed up the software development and delivery process.

Testing: The QA team use tools like Selenium to identify and fix bugs in the new piece of code.

Integration: In this stage, a new functionality is integrated with the prevailing code, and testing takes place. Continuous development is only possible due to continuous integration and testing.

Deployment: In this phase, the deployment process takes place continuously. It is performed in such a manner that any changes made any time in the code, should not affect the functioning of the high-traffic website.

Monitoring: In this phase, the operation team takes care of the inappropriate system behavior or bugs which are found in production.

The following diagram depicts the DevOps lifecycle processes:

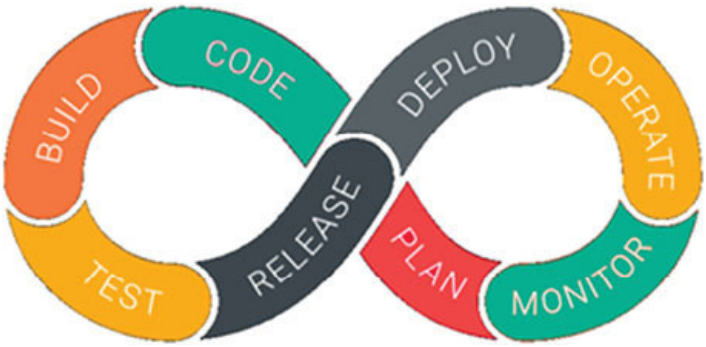


Figure 11.3: DevOps Lifecycle

What challenges did DevOps solve?

Before DevOps, the development and operation team worked in complete isolation.

Testing and Deployment were isolated activities done after design-build. Hence, they consumed more time than the actual build cycles.

Without using DevOps, team members were spending a large amount of their time in testing, deploying, and designing instead of building the project.

Manual code deployment leads to human errors in production.

Coding and operation teams have their separate timelines and are not in sync, thus causing further delays.

There is a demand to increase the rate of software delivery by business stakeholders. As per Forrester Consulting Study, only 17% of teams can use the delivery software fast enough. This proves the pain point.

What are the benefits of DevOps?

DevOps allows Agile development teams to implement Continuous Integration and Continuous Delivery. This helps them to launch products faster into the market. The benefits of DevOps are as follows:

Reduced time to market: DevOps reduces the time to market up to 50% through streamlined software delivery. This is particularly the case for digital and mobile applications. Reduction in time spent in fixing and maintaining applications. Reduction in time spent on development, testing, and operations

Better quality: DevOps helps the team to provide improved quality of application development as it incorporates infrastructure issues.

Predictability: DevOps offers significantly lower failure rate of new releases.

Reproducibility: Version everything so that the earlier version can be restored anytime.

Maintainability: Effortless process of recovery in the event of a new release crashing or disabling the current system.

Reduced risk: DevOps incorporates security aspects in the software delivery lifecycle. It helps in reduction of defects across the lifecycle.

Resiliency: The operational state of the software system is more stable, secure, and changes are auditable.

Cost efficiency and increased revenue: DevOps offers cost efficiency in the software development process, which is always an aspiration of IT companies' management.

DevOps is a proven strategy. Organizations have seen many improvements around business, technology usage and delivery by implementing DevOps.

How is DevOps different from traditional IT?

Let's compare the traditional software waterfall model with DevOps to understand the changes DevOps brings. The following table provides the comparison between Traditional IT and DevOps:

Table 11.1: Traditional Vs DevOps Processes

How is DevOps different from Agile/SDLC?

Agile is a group of values and principles about how to develop software. For example, Agile values and principles are leveraged to build the software in small iterations. DevOps is a way to quickly and easily move the software into production in a safe and flexible way. To do this, one needs DevOps tools and techniques.

The Agile software development methodology focuses on the development of software, but DevOps, on the other hand, is responsible for the development as well as the deployment of the software in the safest and most reliable way.

Which are the top DevOps tools?

The most popularDevOps tools are as follows:

Git: Version Control System tool

Selenium: Continuous Testing tool

Jenkins: Continuous Integration tool

Puppet, Chef, Ansible: Configuration Management and Deployment tool

Docker: Containerization tool

Nagios: Continuous Monitoring tool

The following diagram depicts the process and tools in the DevOps methodology:

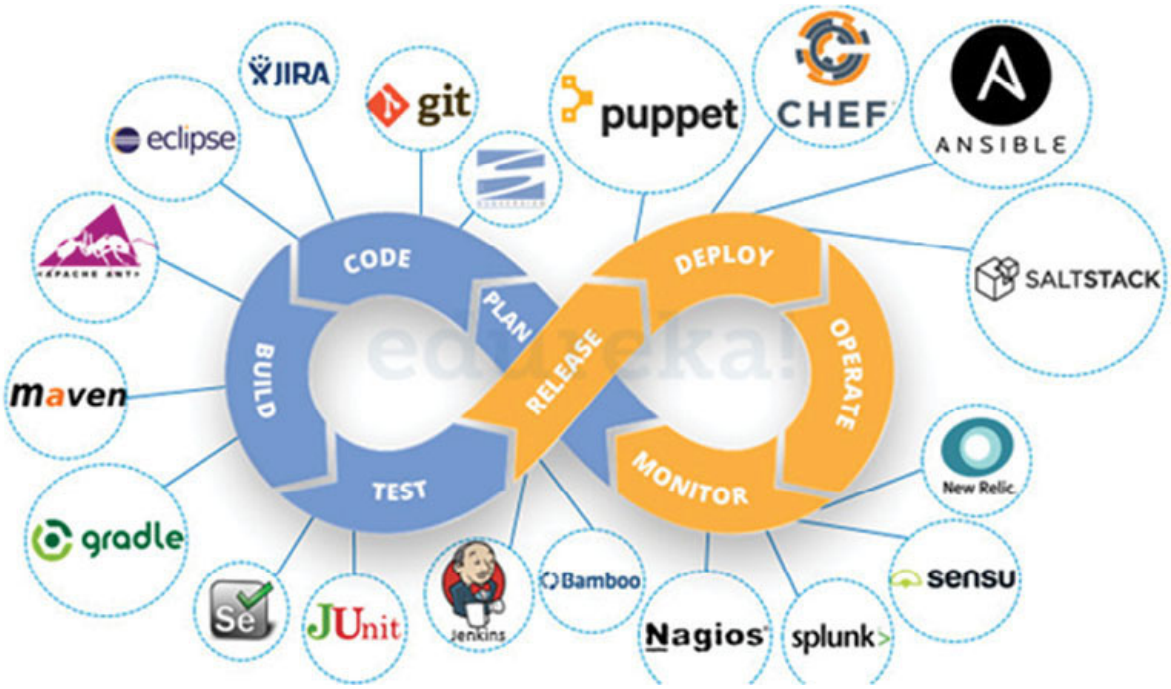


Figure 11.4: DevOps tools

How do DevOps tools work together?

The following is a logical workflow where everything is automated for seamless delivery. However, this flow may vary from organization to organization:

Developers develop the code, and this source code is managed by Version Control System like Git, and so on.

Developers send this code to the Git repository, and then any changes made to the code are committed to the repository.

Jenkins pulls this code from the repository using the Git plug in and builds it by leveraging tools like Maven or Ant.

Configuration management tools like Puppet deploy and provision testing environment and then Jenkins releases this code on the testing environment on which testing is done by leveraging tools like Selenium.

Once the code is tested, Jenkins sends it for deployment on the production server. The production server may also be provisioned and maintained by tools like Puppet.

After deployment, it is continuously monitored by tools like Nagios.

Docker containers provide a testing environment for testing the build features.

Software quality

What is software quality?

Software quality is a meeting requirement, expectation of a customer's needs.

Define Quality Assurance.

Quality Assurance(QA) is the measure of the quality of the process that is leveraged to create a quality application. Quality Assurance is done to measure the quality of the process used to create a better quality application or system. QA makes sure that you are doing the right things, the right way.

What is Quality control?

Software quality control is set of activities for ensuring quality in the software product. Quality control covers variety of reviews such as requirement reviews, design reviews, code reviews, test plan reviews, and test case reviews along with testing activities (Unit testing, Integration testing, System testing, and Acceptance testing).

What is the difference between Quality Assurance and Quality control?

While both Quality control and Quality assurance is part of Quality management, while QC is about detecting/identifying the defect in the work product, QA is about preventing the defect from occurring. Therefore, QA is proactive and QC is reactive.

What is software testing?

Software testing is a process of evaluating an application by manual and/or automated testing to verify that it meets the business requirements. It is a validation process that uncovers the defects in the applications and product. It involves operating the system or application under controlled conditions both normal and abnormal and evaluating the results.

An approach to software testing is to detect the defects of different severity and various priorities at different stages of the software development cycle by leveraging different categories and types of testing.

Explain software testing lifecycle.

The software testing lifecycle consists of the following:

Planning, preparing test strategy, and planning

Developing and creating testing cases and environment

Reviewing the test cases and then updating the same if required

Executing test cases

Result analysis

Bug tracking

Reporting and closure

How do test documents in a project span across the software development lifecycle?

The following figure shows pictorially how test documents span across the software development lifecycle:

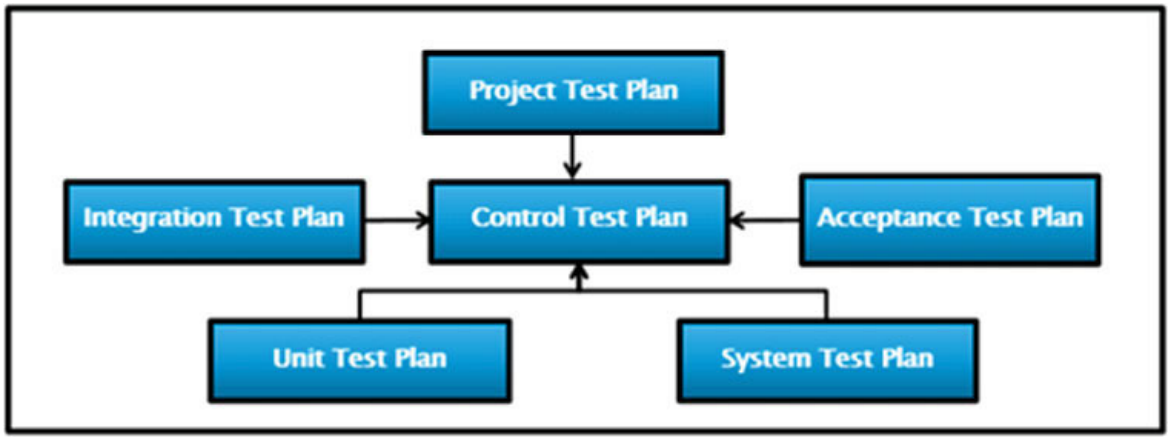


Figure 11.5: Software Development Lifecycle

The following list discusses the specific testing documents in the lifecycle:

Project test plan: This is the main test plan, which outlines the test strategy of the software project. This document should be prepared before the start of the project and should be leveraged until the end of the software development lifecycle.

Acceptance test plan: This test plan is prepared with the end customer. This document commences during the requirement phase and completes at the final delivery.

System test plan: This test plan starts during the design phase and proceeds until the end of the project.

Integration and unit test plan: Both of these test plans start during the execution phase and continue until the final delivery.

On what basis is, the acceptance plan prepared?

The acceptance document is normally prepared using the following inputs:

Requirement document: This document specifies what exactly is needed in the project from the customer's perspective.

Input from a customer: This can be basic discussions, informal talks, emails, and so on.

Project plan: The project plan prepared by the project manager also serves as a good input to finalize the acceptance test.

The following diagram shows the most common inputs leveraged to prepare acceptance test plans:

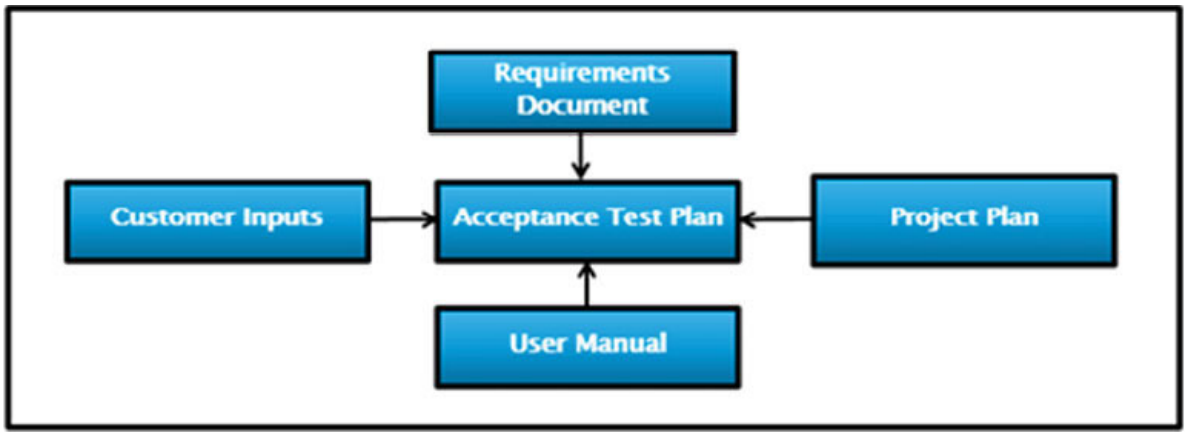


Figure 11.6: Common Inputs for Test Plan

Explain the various types of testing.

Table 11.2: Typical three-tier Architecture

What is the difference between system testing and acceptance testing?

Acceptance testing checks the system against the Requirements . It is similar to system testing in such a way that the whole system is checked, but

the important difference is the change in focus.

The customer knows what is required from the system to deliver business value and is the only person qualified to make that judgment. This testing is more about ensuring that the software is delivered as per the customer needs. It's like getting a green light from the customer that the software meets expectations.

System testing checks whether the system that was specified has been delivered. Acceptance testing checks whether the system will deliver what was requested. The customer should always do acceptance testing.

What is the difference between a white box and black box testing?

Black box: Black box testing is a testing strategy based solely on requirements and specifications. Black box testing requires no knowledge of implementation, structures or internal paths of the software being tested.

White box: White box testing is a testing strategy based on code structures, internal paths, and implementation of the software being tested. White box testing need programming skills.

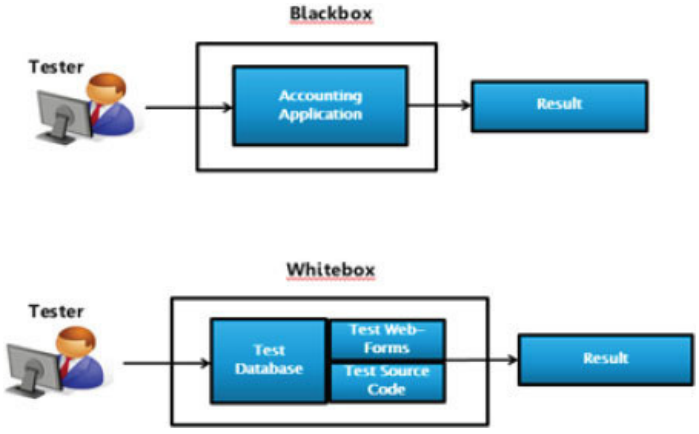


Figure 11.7: Black box vs White box testing

Which test cases are written first: white boxes or black boxes?

Normally, the black box test cases are written first and then the white box test cases. In order to write black box test cases, one needs the requirement document and design or project plan. All these artifacts are easily available at the initial stage of the project. White box test cases cannot be started in the initial phase of the project because they need more architecture clarity, which is not available initially. So normally, white box test cases are written after black box test cases are written.

Black box test cases do not require system understanding, but white box testing needs more structural understanding. Structural understanding is clearer in the later part of the project, that is, while executing or designing. For black box testing, you need to analyze from the functional perspective, which is easily available from the requirement document.

The following diagram depicts black box vs white box testing:

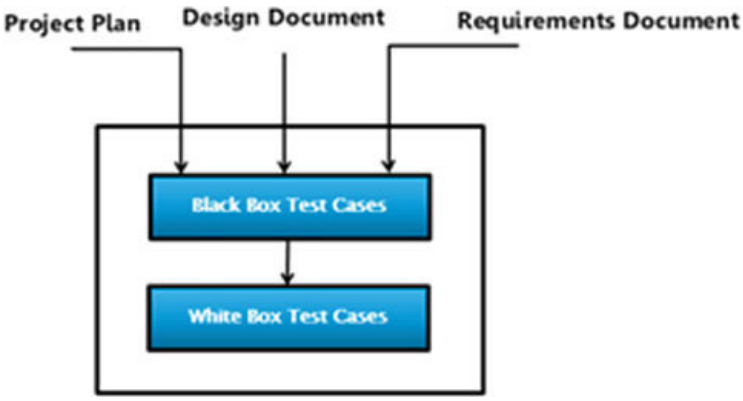


Figure 11.8: Black box vs white box testing

Explain the following terms related to defect management: defect severity and defect priority.

Defect severity: This defines the impact of a defect on an application with respect to the functionality.

Defect priority: This defines the importance of bugs and the urgency to fix the same.

Explain Traceability matrix.

Traceability matrix is used to check the test case coverage with respect to the requirement specification. In this matrix, test cases are traced back to requirements to establish linkage and ensure coverage.

What do entry and exit criteria mean in a project?

Entry and exit criteria are necessary for the success of the project. By defining exit and entry criteria, you define your boundaries. For instance, you can define entry criteria that the customer should provide the required document or acceptance plan. If this entry criterion is not met, then the project will not start. On the other end, you can also define exit criteria for the project. For instance, one of the common exit criteria in projects is that the customer has successfully executed the acceptance test plan.

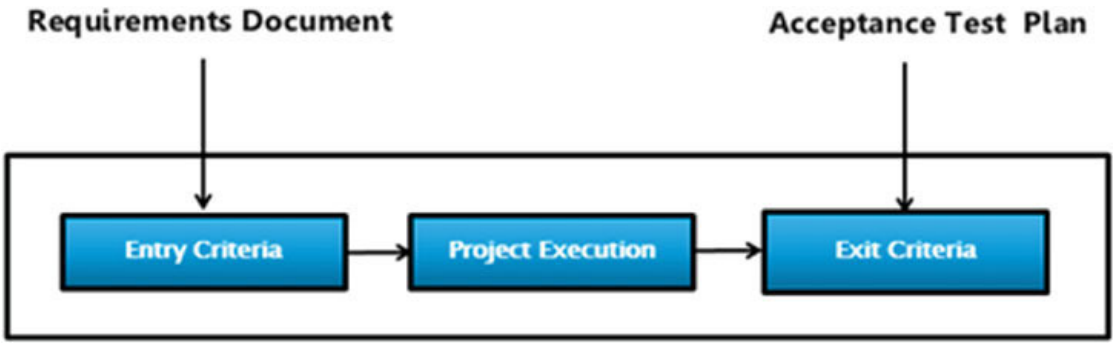


Figure 11.9: Entry exit criterion

A defect, which could have been fixed during the initial stage, is removed at a later stage. How does this affect the cost?

If a defect is identified at the initial stage, then it should be removed during that stage itself rather than at a later stage. It's a known fact that if a defect is delayed for later phases, it proves more costly. The following figure shows how a defect is costly as the phases move forward. A defect that is identified and removed during the requirement or design phase is the most cost effective, while a defect removed during the maintenance phase is 20 times costlier than during the requirement and design phases. The following figure depicts the defect cost basis the phase where it is identified:

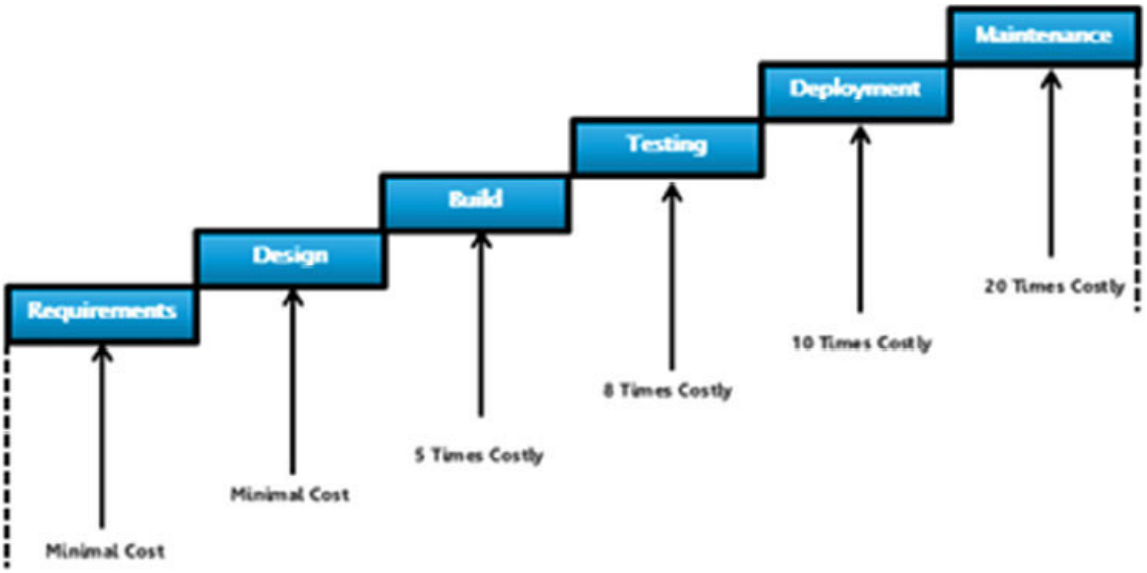


Figure 11.10: Defect cost

For instance, if a defect is identified during the requirement and design phase, one only needs to change the documentation, but if it is identified during the maintenance phase, we not only need to fix the defect, but also change the test plans, do regression testing, and change all the relevant documentation. This is why a defect should be identified/removed in earlier phases, and the testing department should be involved right from the requirement phase and not after the execution phase.

Configuration management

What is version control?

Version control is a system that records changes to a set of files or files over time so that one can recall specific versions. Version control systems consist of a central shared repository where teammates can commit changes for a file or set of files.

Version control allows you to:

Revert files back to a previous state

Compare changes over time

Revert the entire project back to a previous state

Who introduced an issue and when

See who last modified something that might be causing a problem

The following diagram shows the version control system:

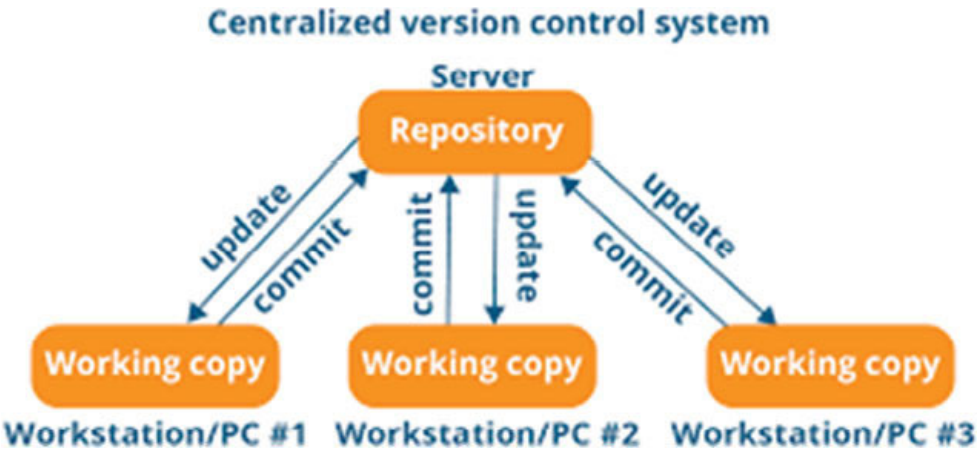


Figure 11.11: Centralized version control

What are the benefits of a version control system?

The benefits of a version control system are as follows:

Version Control System (VCS) allows all the team members to work freely on any file at any time. VCS allows you to merge all the changes into a common version.

All the past versions and their variants are neatly packed inside the VCS. When you need it, you can request any version at any time, and you can get a snapshot of the complete project.

Every time you save a new version of the project, the VCS requires you to provide a short description of changes made to the source files. Additionally, you can see what exactly was changed in the file's content. This allows you to know who has made what changes to the project source files.

A distributed VCS like Git allows all the team members to have the complete history of the project so if there is a breakdown in the central server you can leverage any of the teammate's local Git repositories.

What is meant by Continuous Integration (CI)?

Continuous Integration (CI) is a development methodology that requires developers to integrate code into a shared repository several times in a day. Each check-in is then verified by an automated build, thus allowing teams to detect problems early.

The following diagram depicts CI:

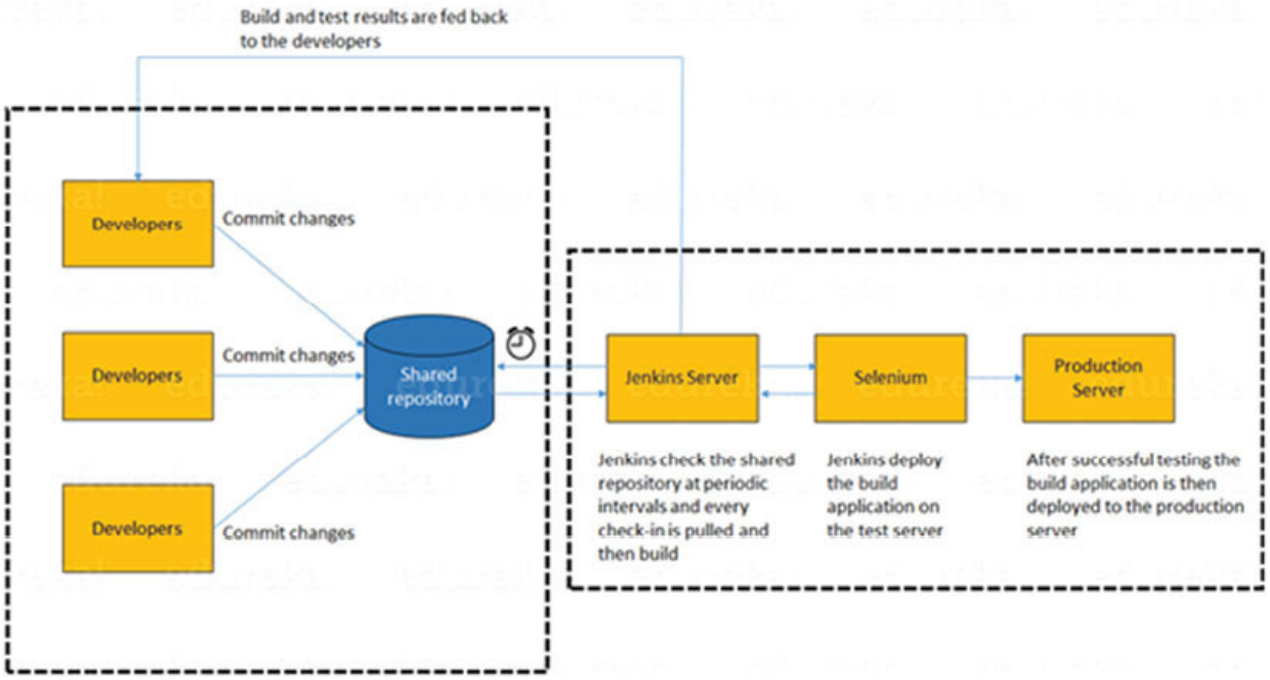


Figure 11.12: Continuous integration

In the preceding diagram:

Developers check out code into their private workspaces.

When they are done, they commit the changes to the shared repository, Version Control Repository.

The CI server monitors the repository and checks out changes when they occur.

The CI server pulls these changes, builds the system, and runs unit and integration tests.

The CI server informs the team of the successful build.

If the build or tests fails, the CI server will alert the team.

The team will try to fix the issue at the earliest.

This process repeats.

Why do you need a Continuous Integration - CI of Dev and Testing?

Continuous Integration of Dev and Testing improves the quality of software and reduces the time taken to deliver it by replacing the traditional practice. It allows the Development team to easily detect and locate problems early because developers need to integrate code into a shared repository several times a day (more frequently). Each check-in is then automatically tested.

What are the success factors for CI?

The following are the success factors for CI:

Maintaining code repository

Automating the build

Make the build self-testing

Everyone commits to the baseline every day

Every commit to the baseline should be built

Keep the build fast

Test in a clone of the production environment

Make it easy to get the latest deliverables

Everyone can see the results of the latest build

Automate deployment

Microservices and SOA

What are microservices?

Microservices is a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services. In themicroservices architecture, services should be fine-grained and the protocols should be lightweight. The benefit of decomposing an application into different smaller services is that it improves modularity and makes the application easier to understand, develop and test. It also parallelizes development by enabling small autonomous teams to develop, deploy, and scale their respective services independently. It also allows the architecture of an individual service to emerge through continuous refactoring. Microservices-based architectures enable continuous delivery and deployment.

Microservice means developing a single, small, meaningful functional feature as a single service. Each service has its own process and communicates with lightweight mechanism, deployed in single or multiple servers. Microservice is an architectural style that structures an application as a collection of self-contained, loosely coupled services. We can create microservices in several languages, including Java, Scala, Node.js, and so on. Here are some of the similarities of microservice with SOA services:

Microservices are loosely coupled and interact with each other through JSON, XML, and so on.

They are reusable and can be accessed through desktop applications, web applications, mobile applications and can be called by other microservices.

WADL, JSON Schema, or Swagger is mostly used for describing the microservices.

Here are some ways MS is different:

There are no physical infrastructure dependencies with microservices. They are usually deployed in Docker containers, which encapsulate both the code and required libraries.

Microservices don’t rely on enterprise products like ESB.

What is a microservices architecture?

A microservices architecture allows you to avoid monolith applications for large systems. It provides loose coupling between collaborating processes, which run independently in different environments with tight cohesion. It is that kind of an architecture that facilitates the avoidance of huge application implementation for a large system. It is associated with the providence of loose coupling that takes place between various collaborating procedures. On the other side, it has the ability to run in an independent manner under various types of situations.

What are the advantages and disadvantages of microservices?

The following are the advantages of microservices:

The smaller the code base it is easy to maintain.

Easy to scale as an individual component.

Technology diversity: We can mix databases libraries, frameworks, and so on.

Fault isolation: A process failure should not bring the whole system down.

Better support for smaller and parallel teams.

Independent deployment.

Deployment time reduce.

The following are the disadvantages of microservices:

ACID transactions do not span multiple processes.

Difficult to achieve strong consistency across services.

Required cultural changes across teams like Dev and Ops working together.

Distributed systems are hard to debug and trace issues.

Greater need for end-to-end testing.

What is monolithic architecture?

A monolithic architecture is like a big container in which all the software components of an application are clubbed inside a single package. The aim of a microservice architecture is to completely decouple application components from one another such that they can be maintained, scaled, and so on. The following diagram depicts a monolithic vs microservices architecture.

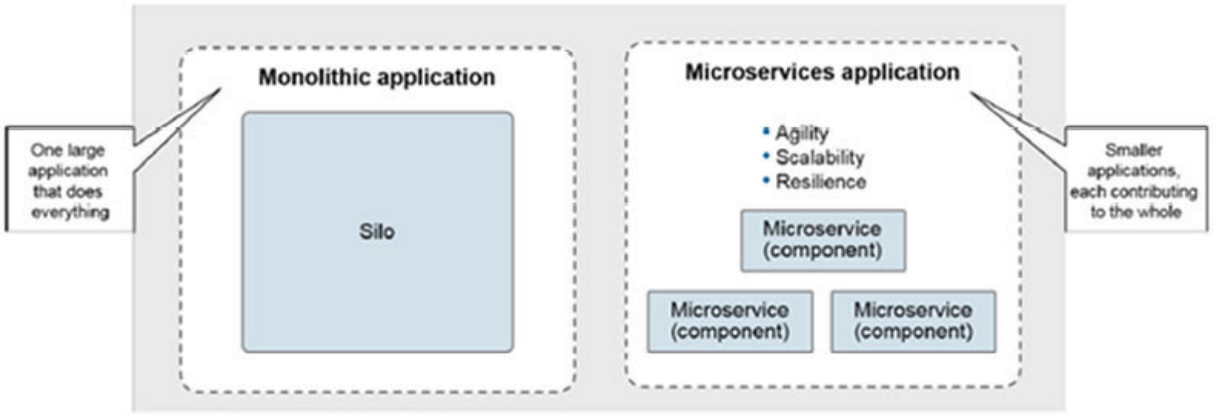


Figure 11.13: Monolith vs microservices architecture

Microservices an evolution of application architecture or SOA

SOA: This is focused on reuse, technical integration issues, and technical APIs.

Microservices: This is focused on functional decomposition, business capabilities, and business APIs.

What are main differences between a microservices and monolithic architecture?

A Microservices architecture:

Service startup is fast.

Microservices are loosely coupled architectures.

Changes done in a single data model does not affect other microservices.

Microservices focuses on products, not projects.

A Monolithic architecture:

Service startup takes time.

A monolithic architecture is mostly tightly coupled.

Any changes made in the data model affect the entire database.

A monolithic architecture emphasizeson the whole project.

The following table compares the monolithic vs microservices architecture:

Category	Monolithic architecture	Microservices architecture
Architecture	Built as a single logical executable	Built as a suite of small services
Modularity	Based on language features	Based on business capabilities
Agility	Changes involve building and deploying a new version of the entire application	Changes can be applied to each service independently
Scaling	Entire application scaled when only one part is the bottleneck	Each service scaled independently when needed
Implementation	Typically entirely developed in one programming language	Each service can be developed in a different programming language
Maintainability	Large code base is intimidating to new developers	Smaller code bases easier to manage
Deployment	Complex deployments with maintenance windows and scheduled downtimes	Simple deployment as each service can be deployed individually, with minimal downtime

Figure 11.14: Typical three-tier Architecture

What are benefits of the microservices architecture over the traditional monolithic architecture?

Microservices-based architectures enable continuous delivery and deployment. Microservices are smaller in size and can be quickly built and delivered. This is a great fit in the Agile development as the business users don't have to wait to see the product. In the Microservice architecture, the individual services can be built in different language like Java and Scala. Also, different microservices could be on different versions of the same language like Java 8 and 9.

Microservices are independent; therefore, one microservice can be independently scaled up or down independent of other microservices. For example, if in an airline ticketing company, the ratio between flight ticket searches and bookings is 50:1, then the search microservice can be scaled up without impacting the ticket booking microservice. This brings down the cost, as the company doesn't have to scale the whole application to meet the performance needs. With microservices, we can easily change the technology or version of individual services rather than having to affect the whole product. Microservices are automation friendly and the changes can be continuously integrated with DevOps to increase speed of delivery.

What are the key differences between SOA and the microservices architecture?

The key differences between SOA and microservices are as follows:

Table 11.3: Difference between SOA vs microservices

What are the differences between containers and VMs?

Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware; containers are more portable and efficient. With the virtualization technology, the package that can be passed around is a virtual machine and it includes an entire operating system and the application. A physical server that is running three virtual machines would have a hypervisor and three separate operating systems that are running on top of it.

By contrast, a server that is running three containerized applications with Docker runs a single operating system, and each container shares the operating system kernel with the other containers. Shared parts of the operating system are read only, while each container has its own mount (for example, a way to access the container) for writing. This means that the containers are much more lightweight and use far fewer resources than virtual machines.

Here is a diagram that compares VMs to containers:

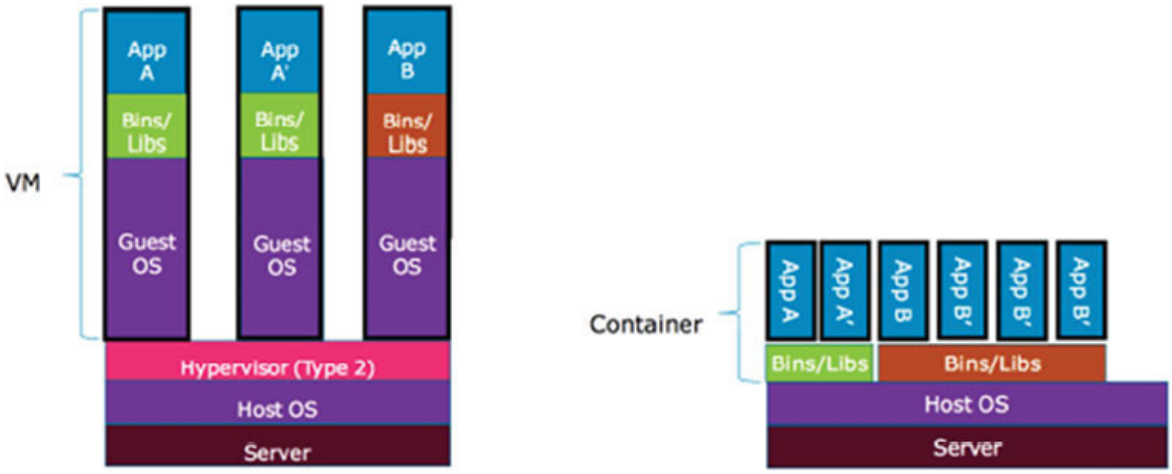


Figure 11.15: Containers vs Virtual Machines

Containers have many of the advantages of virtual machines such as portability. Remember, a virtual machine includes the host operating system, a hypervisor layer, a guest operating system, and the installed software. Unlike VMs, containers do not bundle a full operating system, only libraries and settings that are required to make the software work are needed. This feature makes for efficient, lightweight, self-contained system that guarantees software always runs the same, regardless of where it is deployed. This scenario also converts to a less complicated build process, plus the ability to host many more containers on a single physical server.

A container might be only tens of megabytes in size, whereas a virtual machine with its own entire operating system might be several gigabytes in size. Because of this scenario, a single server can host far more containers than virtual machines. Virtual machines might take several minutes to start their operating systems and run the applications they host; containerized applications can be started almost instantly. Multiple containers can be instantiated while sharing the host operating system as shown on the graphic above. This scenario is useful to respond to increased demand, provide multiple separate profiles, and test newer versions without losing the previous stable version.

What is SOA?

SOA is a set of design principles for building a suite of flexible, interoperable, and reusable services-based architecture. These design principles include discoverable service contract, service abstraction, loose coupling, service reusability, service autonomy, service statelessness, and service composability.

A successful SOA implementation can reduce IT costs by increasing reusability. SOA's flexible mesh of services can also reduce time to market. SOA also helps to leverage existing investments by wrapping legacy applications in a mesh of reusable services. Service-oriented Architecture (SOA) is an architectural style that supports service orientation. Service-orientation is a way of thinking in terms of services and service-based development and the outcomes of services.

What are the key capabilities of SOA?

The following diagram depicts the key capabilities of SOA:

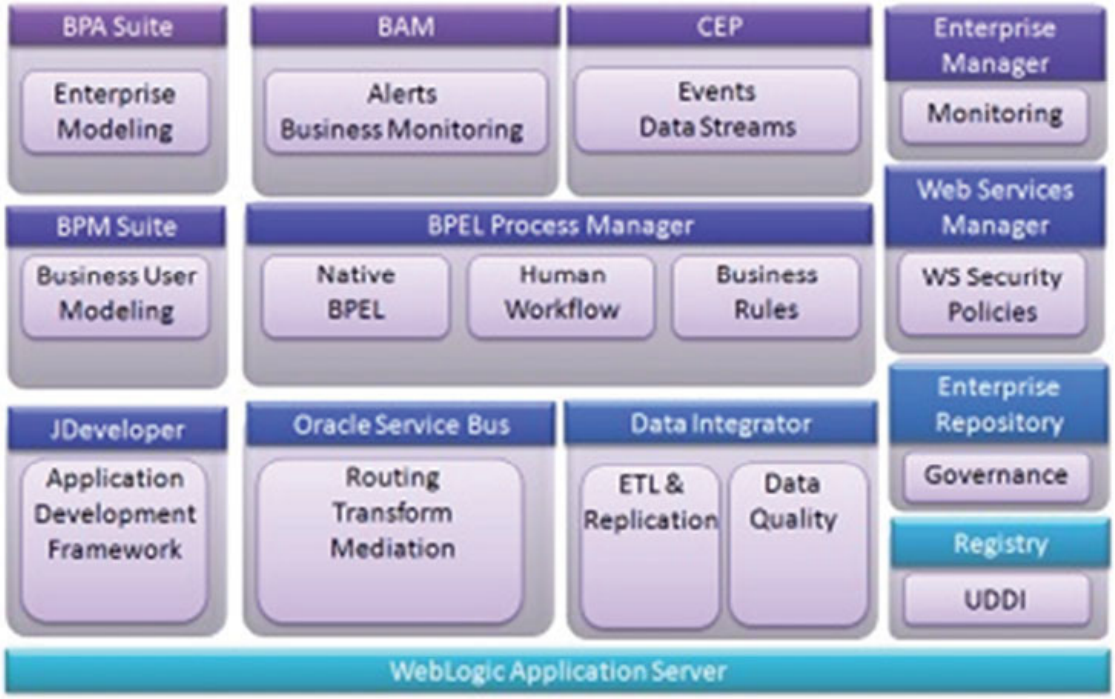


Figure 11.16: SOA capabilities

What are the benefits of leveraging SOA?

SOA helps create greater alignment between IT and line of business while generating more flexibility to support greater business.

Business processes change faster and faster, and global competition requires the flexibility that SOA can provide.

SOA helps get better reuse out of your existing IT investments as well as the new services you're developing.

SOA makes integration of your IT investments easier by making use of well-defined interfaces between services.

SOA also provides an architectural model for integrating business partners', customers' and suppliers' services into an enterprise's business processes. This reduces cost and improves customer satisfaction.

SOA reduces business risk and exposure by helping you comply with proliferating government regulations such as (in the United States) Sarbanes-Oxley, and the US Patriot Act.

Conclusion

This chapter covered an exhaustive question bank that included aspects pertaining to SDLC, software quality, testing, DevOps, Agile, and Scrum. The next chapter will provide guidance on the HR round and will cover topics such as dos and don'ts, HR round objectives, and HR round question bank.

CHAPTER 12

HR Round

The Human resources interview is another short listing stage, where applicants selected from the previous level are weeded away, and the primary goal should be to concentrate on not getting eliminated. Originally, the panel would not really know much about you and the impression one can make in the first few queries is definitely going to establish the precedence for the whole interview conversation and will end up being more or less the deciding factor. The preliminary few minutes are extremely important and therefore, they are the first few queries and their replies.

This chapter will cover the HR round stage of the freshers IT interview process. This chapter presents an exhaustive question bank with special emphasis on practical scenarios and business cases. The chapter is divided into various sections, which includes HR round goals and objectives, question bank, dos and don'ts, and business scenarios. This section also provides the guidelines for adopting key best practices

Goals and objectives

Most companies will evaluate the candidate against the following core parameters:

Communication skills

Candidates believe that this is a language skill, but communication is not about language, or Shakespearean English or the western accent. One needs to have Basic English skills and the three key elements of good communication are as follows:

Listen: Actively listen to the interview panel and do not intrude the question before fully articulating the question.

Articulate: Understand the context and meaning of the question. In case you are unable to, please ask for clarifications. Ask the question to the panel while you understand and get confirmation.

Clear and loud: Once you know the question, answer the question in clear, short and concise words.

Confidence

It's pretty natural for a debutant to get overwhelmed by the environment and the interview panel and to feel nervous. Understand the fact that the organization wants you as much as you need them. They invited you for the HR conversation because they have found the potential in you. Therefore, just be yourself as if you would be in your college or home. A few things that will display confidence are:

Sit straight and keep a comfortable position. Make sure you do not lean or sit on the edge of the seat, nor should you lean back too much.

Maintain a pleasurable smiling expression and show interest in the interview procedures.

Maintain eye contact and don't look at the ceiling while answering the questions.

It's fine to say I don't know . That shows your comfort level in accepting questions you don't have any idea.

Respond with very clear, simple words and loud enough for the panel to hear.

Compatibility with the company

Every organization offers a different culture. When an organization evaluates a potential applicant, one key quality they look for is the compatibility with their organization culture or DNA. Although, one can get the company culture online, preparing for this becomes tricky for tier-2 and tier-3 companies. Here are a few key things you can do to avoid failing:

Display flexibility: Freshers can be groomed to suit the organization's culture. Show them you are versatile and flexible, instead of a tough nut. Keep your choices open like, location, working hours, and so on, unless totally required.

Stay neutral: Do not speakup until your opinion is requested by the panel. They may be evaluating you against one of the company policies.

Stay legal and moral: The organizations policies are based on the moral values and legal code. If the question is what will you do if you see a female colleague harassed by a peer? The answer is I will report it to the HR .

Learnability

Generally, debutant skills in the organization can be flexibly leveraged within different service lines. This could be for a business unit growth roadmap, or helping an existing project, or for product advancements. This will become effective if the learnability of a candidate is strong and steep enough and will be quick and efficient. The learning could end up being a new culture, a new technology or a new lifestyle:

The academic graphs are one of the key process indicators for this and hence, the cut-offs.

This is definitely examined by few indirect questions such as about the experience with teamwork, difficulties you faced and past training.

It will help if you are ready with at least few examples to showcase that you are willing and efficient in learning and adapting new things.

Question bank

Tell us something about yourself.

This is a common question and critical from the perspective of a candidate. Here, the candidates get a chance to talk about self at length. Do not make the mistake of introducing yourself by mentioning the name, qualifications, extra-curricular activities, and so on. The interviewer would like to know details about you, which are not mentioned in the CV. In case you are a fresher, you should start with the value systems inculcated by your parents and family members and then the strengths. Mention why you are the best fit for this job by mentioning the qualifications and the skill sets, which are required to do that job well. Mention them your career plans.

Research the organization website to get all key details. You have to convince the panel that you are keen to make your career with this organization. Before attending the interview, make sure that you are keen to work with this company. Don't look at short-term gains interims of

salaries or attend the interview for gaining experience. The world is a small place and in case you get selected and do not take up the offer, it may go against you.

One might get an opportunity to work with that company, but a few years down the line the person who interviewed you in the past, may turn out to be the boss in another organization and this is where the previous record may go against you. If you have decided to make your career in that organization, your focus should purely be on getting competencies and work experience. Also, appreciate the fact that all tier-1 organizations pay you at par with industry standards. At the beginning of your career, learning is more important than earning.

The answers to these questions set the context for rest of the interview. The good news is that you can earn brownie points if these are answered properly.

Provide a brief summary of your educational history and your family and hobbies. You can also mention your major achievements if it fits the context.

Do not explain strengths and weakness at this point.

Ensure that the answer is in 3-4 sentences and not more.

Ensure that the answers are simple and to the point. Please give time to the panel for further questions. Do not stretch it too long.

A common mistake is that most HR interviews start the conversation with “I am basically from…” This is not the right way to start the conversation.

What are your strengths and weaknesses?

Most candidates do not know about strengths and weaknesses, which are surprising. For example, sales as a function. You need requisite qualities to be an effective sales executive and grow on the job. The qualities are integrity, excellent oral communication and listening skills, excellent knowledge of the product, ability to size up the customer and understand their needs, give the right solution or explain the benefits, high energy levels, patience, discipline, the ability to believe in self when the chips are down. Lastly, believing the product that is the best in the market. The idea is to know how much you are aware about yourself and the confidence in your strengths.

You need to do an honest SWOT analysis of self. You can talk to the family members, friends and ask them to give an honest feedback on the good qualities and weaknesses. The common sense lies in working on one's strength and thereby, raising the level. Coming to weaknesses, keep in mind all professionals have weaknesses. However, you cannot mention a weakness, which is a requisite for the job requirements. For example, in sales, one of the critical qualities of a salesman is his/her ability to speak English fluently. Hence, the candidate should not mention that he/she is weak in oral communication.

Strengths:

Stay positive and even a fundamental answer like I am a very positive individual is good.

You can change this as per the job requirements. Knowing the profile you are being interviewed helps as well.

Explain your strengths with a case study from the past.

Example:

My biggest strength is the ability to learn things quickly. For example, I had to represent my college in an inter-college competition, and I was introduced to the team at the last minute. I was able to articulate our exhibits and artifacts and present them to the audiences with great success.

My best skills would be my articulation skills and verbal abilities. I have worked as an editor for my college magazine and have been involved with various forums during my college and school days.

My greatest strength is my positive attitude, even during the hardest of the times. During our college project, there were times, when our idea was not falling in place, and the results were negative. I had to act positively and keep my team energized to persist on our idea, and finally, we got the desired outcome.

Weaknesses:

This is a tricky question. The expectation here is to mention about a negative trait with a positive twist. Is a weak point of a person, but is all right from an organization perspective.

Example:

I plan to drive multiple things, leaving little time for myself.

I am a perfectionist and hence at times, concentrate too much on one single task to get it to perfection, thus consuming a lot of time.

When someone asks for help, I am working on learning to say no at times.

What is your greatest strength?

This is one of the most commonly asked questions during the interview. The answer to this question demonstrates the preparation for the interview. The same set of strengths may not be an advantage in every interview. You will need to tailor the answer as per the job requirement you are being interviewed for. Before every interview, analyze the role requirements meticulously and articulate the qualities required for the role. Make a list of the qualities you have and match them with the job requirements. You can offer these qualities as your strengths for the role.

What is your greatest achievement?

In case you do not include this part in your profile, the interviewer may ask about your biggest achievements to get a big picture. Therefore, to know who you are, the interviewer would like to assess your career accomplishments so just keep it real. Research something from the past that makes you proud. If you do not find an answer to this question in your professional or academic life, it is fine to offer a reply from your social life.

Example:

My biggest achievement was climbing Mt. Kilimanjaro at the age of 13, and so on.

I managed to save the life of a man hit by a car by taking him to the hospital on time and donating a bottle of blood. This example demonstrates that you are a responsible person.

What is the difference between integrity and honesty?

Integrity means being truthful at a given point of time, even when one is not being observed. Consider that you are the CEO of an organization and your organization has narrowed down on a key supplier, and the proposal has come to your table for approvals. You can easily make money on the deal by finalizing a huge order. A person of integrity will not compromise or give way to any temptation. Honesty means being truthful and fair under all circumstances on the professional and personal front.

What is the difference between character and reputation?

The character is what you are from within, your true self, whereas reputation is what people think about you. Reputation is not at all a true picture of a person as appearances can be deceptive. A dishonest person wears a mask and comes across as a gem of a person. Ultimately, it is your character, which decides the course of your journey.

How would you describe your work style?

You can stress upon your key strengths and mention that your work style is performance driven and you are passionate about your job.

What motivates you at work?

My long-term goals motivate me the most. Like the satisfaction of completing the job, being a part of organization’s success and making my parents proud and happy.

I have always been motivated by the desire to do a good job at whatever position I am in.

Several things are important to be motivated: attention and interest in what I am doing; challenging work that makes my blood running; autonomy and freedom; genuine and open atmosphere/communication where I can work easily and comfortably; respect and recognition both financial and non-financial; keeping promises.

What are your expectations from this first job?

You can always quote a mere vision or strategy, which you like about the organization. As a fresher, your greatest expectation should be To Learn and this is what most organizations expect. There may be additional personal expectations like contributing to society, placing yourself within the corporate world, paying off your education loan, and so on

Examples:

I consider my first job as an extension of my studies and more like a practical class. I want to learn new technologies, learn how the corporate world works, be part of a large energetic team and contribute to projects, develop soft skills and prepare myself for tougher challenges.

I have had mostly theoreticalleanings during the last several years of my graduate curriculum. My first and foremost expectation from my first job would be to learn how all these are applied practically and industrially. I also want to be apart of the corporate culture and find out how this runs.

What are the critical qualities that make a good team player?

Sincerity, dedication, hard work, communication, participation, share, support, and listen.

Are you a team player?

Yes, is what you must say! Keep some examples ready. Mention scenarios or situations where you often perform for the good of the team rather than for oneself. This would be a real evidence of your teamwork attitude. Do not boast about it just say it in a matter-of-fact tone. Keep it to the point. As a fresher, the best would be to describe your efforts as teamwork when you do your projects in academics.

Are you introvert or extrovert?

Being on either of the extremes can be bad for you and the corporation. Your behavior at work should be as per the requirement.

Example:

I behave according to the demands of the situation. For example, if I am representing my organization at an event, which requires me to talk to many stakeholders, I become an extrovert and if there is a serious issue under discussion, I do not jump the gun and interrupt the thought process.

Looking back, what things would you do differently in your life?

Our past is something, which cannot be changed anymore. Ups and downs are part of life, which helps us keep going. I can only learn from my past experiences and try to apply my learning in future.

Why does this role interest you? Alternatively, why have you applied for this job?

I am interested in this job because I believe this opportunity will allow me to use my skills to the best of my ability. I am a hard worker, and also, I love working with people, and I love working in an environment, which is challenging.

What are your hobbies and interests?

You cannot say that you do not have other interests, other than academics. Even if you are studious, the answer must include reading non-textbooks and researching innovative domains in technology. An organization looks for potential candidates who are well rounded and keep varied interests in life. Identify one interest other than academics and prepare answers. Don't include watching TV, eating, sleeping, and so on. Hobby is an activity that is done in our spare time. You need to make very sure that you know everything about the hobby. Let us take two examples; if the hobby is coin collection, you should be aware that the hobby of coin collecting is known as numismatics. If the hobby is listening to music, you need to be precise, as to which kind of music you listen to, as in classical, melodious, etc. Expect many questions on the hobbies. In the second example, you may be asked to name your favorite singer. You can also be invited to name the last album the singer has rendered his voice or the name of his latest album. Never make the mistake of lying as once you get caught, you will cut a sorry figure.

Example:

I like singing, and I also play musical instrument guitar.

I am a numismatist, that is, coin/currency collector. I have been doing it since college days.

In the past, did you take part in extra-curricular activities?

There is a big difference between hobbies and extra-curricular activities. Hobbies are activities, which one does in his/her sparetime. Extra-curricular activities are events at the college or school level where one gets a certificate to backup. It can be a dramatic, sporting event, drawing or singing competition, NSS or NCC, and so on.

Who is your role model?

A role model is a person admired for certain qualities and she/he need not be a celebrity. Even parents, friends, relatives can be role models. Mention all qualities that you admire in the role model. You may not have your role model. In such cases, you can mention that each individual is unique and gifted with certain qualities.

The important part is not who, but why? You should follow it up with why he/she is most inspiring. Tie your response to the job requirements for which you are interviewing. Keep a couple of personas in mind, so you have multiple options based on how the interview is going. Be sure to select professional influences, not parents or partner. Leverage this easy question as an opportunity to continue to bring out the match to the job.

Choosing a hero might is well suited when you would like to discuss that he/she inspired you to overcome adversity on a job. Stress on a quality in your hero that directly relates to a quality needed for the desired position. Share an experience that you used to solve a problem. This is an

excellent opportunity to showcase personal qualities and they will help you excel at the position you are applying for.

Example:

My biggest inspiration is my father. My father has always thought me to least worry about my weaknesses and leverages my strengths. This has developed confidence in me during difficult times.

My greatest inspiration is Sachin Tendulkar. He had a humble upbringing and his dedication and hard work took him to the highest levels in the world of cricket.

The Captain of cricket Mahendra Singh Dhoni is my inspiration! The reason I admire him for is that under the critical situations he is very calm, composed and handles the team very well and with this attitude, he has led our Indian cricket team to victory many times.

What is success for you?

You need to follow your heart. Success need not be related to one's career. For example, for one employee, success could mean dedication and honesty to his job and organization. Likewise, for another person, success could be feeling happy and proud about doing what he or she is doing.

Success is a process of joyfully creating a life, which reflects your highest value and greatest dream. Success is not a destination it is a journey and at this stage, getting a good job and good salary from your company is one of the milestones of this voyage.

How soon can you join us?

Tell them openly if you need time. In case you need to relocate, you should be free from all the commitments. Many candidates in their anxiety commit an earlier joining date, without thinking if it would be possible for them to join the organization. Later on, if you postpone the date, it gives a wrong impression, even before joining the organization. It sends a negative signal that you do not know how to plan schedule and more importantly assert yourself.

Are you ready to relocate?

This is a personal question, and you need to be practical. If the domestic situation demands your presence at your house, obviously you may not be able to relocate. If you can relocate, say for example, within Kerala, be frank and tell the organization about it.

Unless you have any compelling reasons, being a fresher, it is good to be ready for relocation. The IT industry today works on the global delivery model, where offices are based at several locations. They are strategically staffed, as per business requirements, and there could be a need for these skills at a specific location.

Flexibility to suit the needs of an organization is a significant advantage, and since most of freshers travel around with less baggage, it is easier for the organization to relocate them at a short notice. This applies for a foreign country deputation. So denying a relocation means, you are saying no to onsite opportunity.

Example:

I do not see any issue in relocation and it is always a better learning experience than being around with the same team and environment.

Definitely, I believe working at overseas locations certainly helps individuals to learn new culture and people.

Are you open to working in shifts?

Employees have a false notion that working in shifts leads to health challenges. Working in shifts is nothing new and has been there for several years. This question checks your flexibility. If there is a genuine problem to work in shifts, mention that to the panel.

Are you willing to change your project and profile as needed by the business requirements?

As a fresher, the most sought after quality is flexibility. You need to showcase, flexibility and should be willing to learn. Mentioning you are flexible doesn’t necessarily mean that you will be forced into things, you do not like doing. It just shows the positive attitude to learn and take up new things.

Example:

As a fresher, I am keen to learn as much as possible. I believe playing different roles and profiles only help me learn and grow in this industry.

I understand it will be in the best interest of the organization if my manager wants me to develop new skills and take up another role.

How long will you work for us?

Every employee will aspire for a change at some point in their career. If the organization you are going to join has an excellent reputation and a good work ethic and culture, there is no need to change the job, assuming your performance is consistent. When an organization recruits and trains an employee, they invest money in that person. They expect the employee to be with the organization and make a significant contribution towards the growth of the organization.

Where do you see your-self in five years from now?

This question is intended to know your career ambitions, and how long would you be committed to the organization and are there any immediate plans to move on. In case there is an actual plan, explain the reason. If you do not have any specific plans, give a generic answer along with some interesting activity you would like to take up.

Example:

As a fresher, the first five years will be a learning curve. However, I would like to articulate the industry knowledge and sharpen my technical abilities.

I see myself becoming a solution architect in next five years. My core competency is technologies, and I would like to work and contribute in the field of most recent technologies innovations.

Have you been in some challenging situation?

This is an add-on question to discuss one's strengths mentioned earlier. This is a good opportunity to mention a suitable example such as to mention your strengths.

Example:

We did have a challenging situation during our final semester project. The code we were trying to implement, worked with a 64-bit processor, while we were trying to make it collaborate with a different configuration, due to budgetary constraints. While it was possible the oretically, the result was not coming through, and the team wanted to upgrade the hardware. I was sure, that this will work and that we stick to our original plans and resolve the actual problem. It took the effort to convince the team, a lot of hard work and extra hours of reading, but we stayed on track and cracked the issue and delivered the project.

The preceding example explains strength as positive thinking, hardwork, and so on.

I have a passion for sports and am a frequent member of the college athletic group. There was a competition just before our internal assessment; I was sure I would be able to win points for my team. I decided to put extra hours in my studies, while also practicing and attending the event. I had to stretch and manage my time efficiently. However, finally, I was able to win the championship and come out with flying colors in the exam.

The preceding example shows your passion, multitasking skills, time management, and hard work.

Why should we hire you?

As a fresher, you cannot commit to expertise in any particular domain. Therefore, this question is intended to know the quality of yours that will be critical for the organizations. Research about the company and job qualities required for the organization and this will be the key to this response. In case you do not know about the company flexibility, quick learning ability, and team player are a few key qualities that will fit many organizational needs.

Example:

I had the industry exposure and internships in the past. Being a fresher, it will be flexible to adapt to the environment and reduce the training budget.

I have a sharp learning curve, which will assist ramping up quickly as a productive resource. An example of this is my final year project, where I worked on an entirely new technology in a short time frame.

I am a good leader and team player. I have led teams for cultural and technical activities during my college and school days. I am team player within the challenging team and provide leadership when any opportunity comes up.

I have a unique combination of strong technical competencies, and the ability to build long lasting relationships.

I can be an asset to any organization since I am dedicated towards my work. I have the patience to hold up the pressure and work sincerely.

I have a quality to learn from my failure and don’t give up until I achieve results.

What are your future goals?

My short-term goals (examples):

As a college graduate, I need to start building a significant presence in the industry, working for a company I respect and doing a job that I enjoy.

I want to earn a job, which gives me job satisfaction, and give me a platform where we can excel my skills and knowledge.

My long-term goals (examples):

To achieve a higher esteemed position in the company and be a high performing individual whom the company does not want to lose at any cost.

Long-term goal is to prove myself in every challenging responsibility and be a role model for others.

Do you plan to pursue higher education?

If you are seriously considering graduation, it is wise to say so.

If you are unsure, the answer could be: I am interested in advanced studies, but not immediately. I want to get working and then look for higher education based on my career progression and experience.

Yes, but not immediately. I plan to take up part-time studies, either MBA or masters in engineering, depending on which will be beneficial for my areas of competencies. Some companies support and provide various incentives for part-time and full-time studies.

What motivates you to do a great job?

This is a personal trait that you have to analyze and mention to the interviewer.

Examples: Challenges, achievements, and recognition.

What was the toughest decision you have ever made?

To me every decision depends on the situation. I always believe in taking an appropriate decision in any given situation and then prove it right.

How could you improve your career progression?

Hard work and dedication are two key weapons of a career. If we apply those at appropriate times, then definitely your career turns as you wish. Career progress depends on your profile. If I am ready to take on challenges, learn new skills and accept targets, automatically, my career progress will improve.

What will you do if you are offered a job with a salary higher than ours?

HR wants the candidate to stay with the company for a longer period. The recruitment process is time consuming and costly. Organizations will not hire a candidate who is willing to switch jobs for a few thousands more.

Example:

I am a fresher, and if I switch the jobs frequently, then it will be a question mark on my stability and will affect my career. I have to learn and increase my knowledge base as it benefits if you are knowledgeable. I am here to learn and provide my very best to the organization.

I know it is attraction of money but at this point, growth is more important. In such a scenario, I will discuss growth opportunities with my supervisor and analyze the situation. If there is a growth potential in the same organization, I will continue to work. However, if I do not get further growth opportunities, I will humbly inform about my decision to move on from the organization.

Have you ever worked under pressure? Give an example.

Pressure is an inseparable part of work. At times, it will be the pressure on enhancing the performance, or it could be the pressure of meeting deadlines. The objective of this question is to validate and how you handle pressures. Mainly, emphasize on what you did to not let your focus dilute like prioritizing right activities, creating a proper plan to meet the expectations, starting, and finishing them as per the roadmap and staying dedicated to the outcome.

Example:

Pressure is something, which takes the best out of me, and one delivers the best not just to do what you were asked to but also deliver more than expected.

If you think work should be done in a particular way while the others believe that it should be done in some different way, how would you resolve

the conflict?

No doubt, there can be many approaches to do work. It is a key that you evaluate each of them and take the right one.

Example:

You should understand that there is more than one approach to accomplish things. If I am in a situation, I will evaluate all approaches and suggest the one that is right and get best results.

The purpose here is to check your level of acceptance and adamancy.

What contributions could you make to this organization that would help you stand out from other applicants?

There are a few key steps that can assist in preparing you for an appropriate statement to communicate that you standout:

Understand the role requirements and the company profile. Research about the key deliverables expected from the employee.

Make a list of strengths. The strengths need to include your transferable skills such as communication, interpersonal-relationship skills, time management, negotiation, and soft skills. Analyze the strengths with the keyskills that the panel might be looking for in an applicant. If there are skill gaps, plan to answer in such a way that you address the concern.

Communicate your motivation level. Employers love motivated employees. There is one critical point to remember that you need to be assertive while answering for making the interviewer realize that you are the right fit for the job.

What have you done to improve your knowledge in the last year?

This question checks a candidate's willingness to learn and keep oneself up-to date with ever changing technologies.

Example:

I enrolled myself in several training and certification courses useful for the next stage of our current project. I attended seminars and conferences on personal development and managerial skills improvement.

How do you propose to compensate for your lack of experience?

I know that many companies in this industry require years of engagement experience. However, there are benefits of being a fresher. I am very passionate about my work. Everything is new to me, so I am that much more excited to learn new things.

What are you looking to learn from this organization?

The answer will really depend on where you are on the career path. If you are new, then you may want to gain more experience and learn a wide variety of things. If you have experience, you ought to find challenging assignments and more opportunities for growth.

Why do you think that this company will be a career for you rather than a job?

As I am a fresher, I would like to build up a career rather than money or a job. Every year this company has been the platform for many fresh graduates like me. Many employees have been benefited through this company. So, I consider this as a good opportunity to build my career and future.

Why do you want to work for us?

Mention good points about the company and at the same time, try to align your strengths with the working culture or organization goals.

Example:

Life is a mix of hard work and fun. I find the work culture of this company to carry the same principle. Work hard… Party harder . What better way to start a career than at a place, where I will feel like home.

Working for an organization that has won the best employer award multiple times in itself is a motivating factor. However, more importantly, I would like to start my career in an organization, which values ethical HR practices and encourages the overall development of the employees.

I have always admired this organization for the cutting edge products and developments it brings about in the modern age. The impact of the work is directly reflected in the society in general, which is a great feeling. I would like to be a part of this modern day game changer bandwagon.

What is one thing that you like about our company?

Always research about the organization before choosing the answer. A wrong answer may cost you the job. For example, you go to a research organization and tell about your partying nature, you most likely lose the chance to get through.

How much salary do you expect?

As a fresher, you will usually not have much bargaining power regarding the salary. You can still ask the interviewer about the package they would offer to a fresher. Most of the big organizations have a fixed salary for each level, and there will not be room for much negotiation. However, if you are required to spell out numbers, offer a range rather than an exact number.

Do you want to ask us something about the company?

Not asking any questions demonstrates lack of interest in the job, lack of research or dullness about the company. When you are asked this question, you can ask some interesting questions like the opportunities for you to grow in this current role and the expansion plans of the company. If the company has been in the news recently, you can ask questions related to that also. For example:

Does the company allow for lateral and vertical role changes

If the company encourages learning and development of employees

Expansion plans

You can discuss the role in detail

Things never to say during an interview

I don't really like or enjoy doing my current assignment/job.

There is no learning in what I am doing currently.

I have a vacation planned in a few weeks. (You will have to wait to ask for time-off until you have a job offer).

How is the salary package? (Let the employer talk about it first.)

What is included in the benefits package? (Wait for a job offer before discussing benefits.)

What is the annual leave policy?

When do I get a vacation? (Don't ask about benefits until you're offered the job.)

Can you give me taxi fare to get home? (Figure out your transportation ahead of time.)

Do you mind if I take this call? (Your phone should be turned off before you head into the interview.)

I don't have all the experience you need, but I'm a quick learner. (Let the interviewer figure out if you're qualified and focus on the skills that you do have.)

I have not studied this language as part of my curriculum or project.

I don't know.

It's already mentioned on my resume. (Yes, it is but the interviewer wants to hear it from you.)

I have an appointment, is this going to be over soon? (Give yourself plenty of time to interview and be aware the interview could run longer than you planned.)

Sorry, I'm late. (Don't be unless you have an emergency.)

Profanity or swear words. (Keep it professional and polite.)

This schedule doesn't really work for me. Can it be changed? (Don't ask for anything until you have a job offer.)

I really need this job. (You don't want to come across as desperate.)

I don't have any questions. (You should always have a list of questions ready to ask the interviewer.)

Interviews make me really nervous. (The interviewer wants to hire someone confident in his or her abilities.)

Can I work from home? (Don’t bring up alternative working situations until you have a job offer.)

Conclusion

This book will help aspirants prepare for the IT interview process while emphasizing on the importance of sufficient preparation. With this practical hands-on guide, the readers will not only get their hands onto industry standard IT interview practices and tips, but will also get curated, situation-specific and timeline-specific interview preparation techniques, which will help them take a leap ahead of others in the queue. This book has provided readers with sample questions, which are asked by the top IT companies while hiring. They will get a similar set of questions, which might arise as a corollary from those, which are already asked. They will get hints on solving them as you move ahead and each hint will be customized in a way your actual interview will progress. Whether they are planning to prepare for an interview through a semester for six months or preparing for just a weekend coding competition, this book will have all the necessary information that will lead you to your first successful job. All the best!

Index

Symbols

include file

include file 98

difference between 187