

IBM COBOL Software Development Process

Welcome to IBM COBOL Software Development Process! By enrolling in this course, you are taking a big step in increasing your knowledge and hands on experience with COBOL, IBM z/OS, and the software development process. In this course, you will be introduced to core concepts and technologies associated with COBOL, IBM z/OS, and the software development life cycle (SDLC). You will learn about life cycle and systems analysis concepts. We will also go through design and structured coding. This course covers a wide variety of topics that are designed to provide relevant, requisite working knowledge and practical hands-on experience.

This course is comprised of the following 5 modules:

1	Module 1 - Welcome to COBOL Software Development Process	5
1.1	Course Objectives	5
1.2	Class Introductions	5
1.3	Additional Course Details	5
2	Module 2- Lifecycle Concepts	5
2.1	Software Development Lifecycle (SDLC)	5
•	SDLC fundamentals	6
•	SDLC phases	7
•	DevOps on mainframe	7
2.2	SDLC in Your Reading	5
•	Writing a program	7
•	Compiling	7
•	Executing	7
•	Program logic	7
•	Structured programming	7
2.2.1	Reading link: http://www.pgrocer.net/Cis12/cobollan.html	8
2.3	Steps in a SDLC	5
2.4	Steps in a SDLC in Your Reading	5
•	Design phase	8
•	Gather requirements	8
•	Code and test application	8
•	Perform user tests	8
2.4.1	Reading link: https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-application-development-life-cycle-overview	8
2.5	Research Steps in a SDLC and Share with the Group	5
2.6	Agile Methodologies	5
•	What is Agile?	9

•	Keys to Agile Transformation _____	9
2.7	Agile Methodologies in Your Reading _____	5
•	Tools _____	10
•	Strategies _____	10
2.7.1	Reading link: https://techchannel.com/01/2020/devops-ibm-z-long-way _____	10
2.8	Research Agile Methodologies and Share with the Group _____	5
3	Module 3 - Systems Analysis Concepts _____	5
3.1	IBM Systems Architecture _____	5
•	Expose core mainframe assets _____	11
•	Modernize DevOps _____	11
•	Develop and deploy a new cloud workload _____	11
•	Transform core application and data assets _____	11
3.2	IBM Systems Architecture in Your Reading _____	5
•	Hardware goals _____	11
•	A tour of the central processor complex (CPC) _____	11
•	Input/output operations _____	11
•	Configuration scenarios _____	11
3.2.1	Reading link: http://www.redbooks.ibm.com/redpapers/pdfs/redp5346.pdf _____	12
3.3	Research IBM Systems Architecture and Share with the Group _____	6
3.4	Objectives of Systems Analysis _____	6
•	Definition of overall system processing _____	12
•	Identification of missing and duplicate components _____	12
•	Listing of areas for special attention during conversion _____	12
•	Documentation of all areas of concern _____	12
•	Summary of primary findings _____	12
•	Presentation of recommended actions _____	12
•	Another word on documentation _____	12
3.5	Objectives of Systems Analysis in Your Reading _____	6
•	IBM Operations Analytics – Log Analysis _____	12
•	IBM z Advanced Workload Analysis Reporter (IBM zAware) _____	12
3.5.1	Reading link: https://www.ibm.com/docs/ja/zoa/4.1.0?topic=platform-component-overview _____	13
3.6	Research Objectives of Systems Analysis and Share with the Group _____	6
3.7	Core Areas of Analysis _____	6
•	Vendor _____	13
•	Product _____	13
•	Execution platform _____	13
•	Cover measurement _____	13
•	Data flow _____	13
•	Code execution _____	13
3.8	Core Areas of Analysis in Your Reading _____	6
•	Targeting performance critical sections _____	13

• Optimizing	13
• Migrating	13
3.8.1 Reading link: https://www.ibm.com/support/pages/node/738777	14
3.9 Research Core Areas of Analysis and Share with the Group	6
4 Module 4 - Design and Structured Coding	7
4.1 Pseudocode	7
• Developed to increase readability and modifiability,	15
• Resolves machine deficiencies (e.g. no indexing or floating point)	15
• Expanded to machine code by Grace Hopper with the UNIVAC compiling system	15
○ A note on Grace Hopper	15
4.2 Pseudocode in Your Reading	7
• How to compile and run a program	15
• Understanding the program structure	15
• Resources for mainframe programming	15
4.2.1 Reading link: https://medium.com/@yvanscher/7-cobol-examples-with-explanations-ae1784b4d576	16
4.3 Research Pseudocode and Share with the Group	16
4.4 Modern Design	7
• Replacing existing systems	16
• Modernizing COBOL code	16
• Challenges	16
4.5 Modern Design in Your Reading	7
• Current demands for COBOL	16
• Why you should learn COBOL	16
• How to learn COBOL	16
4.5.1 Reading link: https://medium.com/swlh/why-you-should-learn-cobol-in-2020-9b263bee775d	17
4.6 Research Modern Design and Share with the Group	7
4.7 Planning and Unit Testing	7
• Prepare your testing environment.	17
• Fix your COBOL programs.	17
• Start unit testing.	17
• Invest in source debuggers, utilize compile options.	17
4.8 Planning and Unit Testing in Your Reading	7
• Classification of manual testing in mainframe	17
• How to do mainframe testing	17
• Mainframe automation testing tools	17
4.8.1 Reading link: https://www.guru99.com/mainframe-testing.html	17
4.9 Research Planning and Unit Testing and Share with the Group	7
4.10 Top-bottom Testing	7
• COBOL is executed top-down structures.	18

•	Top-down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.	18
•	Sandwich testing is an approach to combine top-down testing with bottom-up testing.	18
4.11	<i>Top-bottom Testing in Your Reading</i>	7
•	Definition of top-down approach	18
•	Definition of bottom-up approach	18
•	Key differences between top-down and bottom-up approach	18
4.11.1	Reading link: https://techdifferences.com/difference-between-top-down-and-bottom-up-approach.html	18
4.12	Research Top-Bottom Testing and Share with the Group	7
5	<i>Module 5 - Congratulations on Completing COBOL Software Development Process</i>	7
5.1	Completed Course Objectives	7
5.2	Final Class Comments	7
5.3	Additional Course Details and Next Steps	7

1 MODULE 1 - WELCOME TO COBOL SOFTWARE DEVELOPMENT PROCESS

1.1 COURSE OBJECTIVES

1.2 CLASS INTRODUCTIONS

1.3 ADDITIONAL COURSE DETAILS

2 MODULE 2- LIFECYCLE CONCEPTS

2.1 SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)

2.2 *SDLC* IN YOUR READING

2.3 STEPS IN A SDLC

2.4 *STEPS IN A SDLC* IN YOUR READING

2.5 RESEARCH STEPS IN A SDLC AND SHARE WITH THE GROUP

2.6 AGILE METHODOLOGIES

2.7 *AGILE METHODOLOGIES* IN YOUR READING

2.8 RESEARCH AGILE METHODOLOGIES AND SHARE WITH THE GROUP

3 MODULE 3 - SYSTEMS ANALYSIS CONCEPTS

3.1 IBM SYSTEMS ARCHITECTURE

3.2 *IBM SYSTEMS ARCHITECTURE* IN YOUR READING

3.3 RESEARCH IBM SYSTEMS ARCHITECTURE AND SHARE WITH THE GROUP

3.4 OBJECTIVES OF SYSTEMS ANALYSIS

3.5 *OBJECTIVES OF SYSTEMS ANALYSIS* IN YOUR READING

3.6 RESEARCH OBJECTIVES OF SYSTEMS ANALYSIS AND SHARE WITH THE GROUP

3.7 CORE AREAS OF ANALYSIS

3.8 *CORE AREAS OF ANALYSIS* IN YOUR READING

3.9 RESEARCH CORE AREAS OF ANALYSIS AND SHARE WITH THE GROUP

4 MODULE 4 - DESIGN AND STRUCTURED CODING

4.1 PSEUDOCODE

4.2 *PSEUDOCODE* IN YOUR READING

4.3 MODERN DESIGN

4.4 *MODERN DESIGN* IN YOUR READING

4.5 RESEARCH MODERN DESIGN AND SHARE WITH THE GROUP

4.6 PLANNING AND UNIT TESTING

4.7 *PLANNING AND UNIT TESTING* IN YOUR READING

4.8 RESEARCH PLANNING AND UNIT TESTING AND SHARE WITH THE GROUP

4.9 TOP-BOTTOM TESTING

4.10 *TOP-BOTTOM TESTING* IN YOUR READING

4.11 RESEARCH TOP-BOTTOM TESTING AND SHARE WITH THE GROUP

5 MODULE 5 - CONGRATULATIONS ON COMPLETING COBOL SOFTWARE DEVELOPMENT PROCESS

5.1 COMPLETED COURSE OBJECTIVES

5.2 FINAL CLASS COMMENTS

5.3 ADDITIONAL COURSE DETAILS AND NEXT STEPS