Ministry of Education

Test Blueprint for National Exit Examination to be held in 2015 E.C

Bachelor of Science Degree in Computer Science

Compiled By:

Shumet Tadesse, shumet.nigatu@gmail.com, University of Gondar

January, 2023

Bahir Dar

Ethiopia

Table of Contents

List of Tables

1. Introduction

The Ministry of Education of Ethiopia has announced the implementation of exit exam for all undergraduate program students (public and private), beginning with the 2022/2023 academic year, in order to improve the quality of graduates produced by higher learning institutions. The exit exam aimed at checking whether students have acquired the required knowledge, skills and attitudes or not. To implement this, it requires determining competency areas for a specific program, which is already completed. Based on the competency areas prepared, it needs to plan the construction of tests.

Planning of a test is a first and vital step in the construction of an achievement test. An achievement test demands very systematic and careful planning, as a fact that good planning is a symbol of success. Tests are the tools that provide scores that measure level of student learning and study program learning outcomes. In order to achieve the valid and reliable measurement of student learning and program learning outcomes, the development of valid and reliable test is the mandatory. Test should be able to measure student performance in all dimensions of knowledge, skill and attitude. The carefully planned test construction contributes to improve the overall quality of the test in terms of test content validity, difficulty level, discrimination power and test reliability. Test preparation is not an easy task; it requires a careful planning and guideline to make the task simple. Test construction needs the preparation of test blue print. Test blueprint is defined as a complete plan that explains how to develop a test. The term refers to a map or specification of assessment to ensure that all aspects of the curriculum and educational domains are covered by the assessment programs over a specified period of time. It helps curriculum developers/test constructors to match various competencies with the course content and the appropriate modality of assessment.

Generally, test blueprint will help to ensure tests: 1) Appropriately assess the achievement of instructional objectives of the course; 2) Appropriately reflect key course goals, objectives and the material learned or covered during the instruction period; and 3) Include the appropriate item formats along with the knowledge and skills being assessed.

Keeping this in mind, the team has prepared this test blueprint document in order to help the test developers or content specialists in their process of valid and reliable test construction. The major points considered in the process of preparing this test blue print guideline were the core competencies that have been already identified for the themes of courses, the course contents, course credit hours, and the learning outcomes with their corresponding levels of achievement by learning domains. In line with this, the

primary goal of this blueprint is to determine the number of test items across themes and courses, as well as the mapping between learning outcomes and skill sets for the computer science BSc program.

2. Objective of test blueprint

Test blueprint preparation is generally opted to assist the preparation of a test that is representative, broadly sampled, and consisting of complete knowledge domain expected of the Ethiopian higher education students on completion of their study program. The specific objectives of test blueprint are to:

- Facilitate the construction of a representative and balanced test items for the selected courses in accordance with the competencies identified.
- Guide test developers or writers to write or set appropriate test items.

3. Expected profiles of graduates

The expected graduate profile of computer science graduates is outlined below in terms of the three basic metrics mentioned above.

3.1. Knowledge

Graduates of computer science are expected to understand both the theoretical and practical aspects of field, as well as the role of computing systems in general. To this end, the ability to apply or justify concepts, methods, and computational proficiency in the field is required. It is, therefore, critical to have comprehensive knowledge and understanding of the following topics:

- The fundamental concepts, principles and theories of computation and the application of computers.
- Software Fundamentals and programming languages
- Systems architecture and infrastructure
- Systems modeling
- Structuring of data and information
- Hardware
- Trends and developments in computer science

3.2. Skills

The following skills are also expected from computer science graduates:

- Remembering emerging technologies
- Understanding computer architecture and operating systems

- Applying programming languages and software tools to address issues in the real world
- Analyzing existing computing infrastructures and architectures
- Evaluating systems in terms of general quality attributes and potential tradeoffs
- Creating computer artifacts to solve societal problems by applying system modeling, development, and implementation principles

3.3. Attitudes

Graduates of computer science are expected to have a wide range of transferable skills (attitudes), including

- Teamwork: Capable of making a valuable contribution to a development team.
- Communication: Briefly explain technical problems and how to solve them to a range of audiences.
- Handling Ethical Issues in Computer Technology: Recognize and follow the social, professional, and ethical issues that arise from the use of computer technology.

4. Themes and list of courses

Table 4-1: Courses organized into themes

Theme	Courses
System Development	Software Engineering
	Web Programming
	Database Systems[1]
Programming and Algorithms	Computer Programming
	Object Oriented Programming
	Design and Analysis of Algorithms
	Data Structure and Algorithms
Computer Networking and Security	Data Communication and Computer Networking
	Computer Security
	Network and System Administration
Intelligent Systems	Introduction to Artificial Intelligence
Computer Architecture and Operating Systems	Operating System
	Computer organization and architecture
Compiler and Complexity	Automata and Complexity Theory
	Compiler Design

[1] Fundamentals of Database Systems and Advanced Database Systems

5. Total credit hour of the selected courses

Table 5-1: Courses chosen and their credit hours

No	Course Name	Credit Hour
1	Computer Programming	3
2	Database Systems[2]	6
3	Object Oriented Programming	3
4	Computer organization and Architecture	3
5	Data Communication and Computer Networking	3
6	Data Structures and Algorithms	3
7	Web programming	4
8	Operating System	3
9	Software Engineering	3
10	Design and Analysis of Algorithms	3
11	Introduction to Artificial Intelligence	3
12	Computer Security	3
13	Network and System Administration	3
14	Automata and Complexity Theory	3
15	Compiler Design	3
Total		**49**

6. Test Blueprint

Ministry of Education
Higher Education Sub-sector
Test Blueprint (Table Specification) for BSC in Computer Science
Table 6-1: Share of themes, courses and items of a course from a total of 100 test items

Themes	Course Name	Credit hour	Weight of course or proportion	Number of test items from each course	**Cognitive**						**Affective**	**Psychomotor**	Total
					Remember	Understand	Apply	Analyze	Evaluate	Create			
System Development Share = 27 items from the total	Software Engineering	3	3/13 = 0.23	0.23*27 = 6	-	2	1	1	1	1	-	-	6
	Web Programming	4	4/13 = 0.31	0.31*27 = 9	-	2	2	2	2	1	-	-	9
	Fundamental of Database Systems	3	3/13 = 0.23	0.23*27 = 6	-	2	1	1	-	2	-	-	6
	Advance Database Systems	3	3/13 = 0.23	0.23*27 = 6	-	1	2	-	1	2	-	-	6
	Total credit	**13**											
Programming and Algorithms Share = 25 items from the total	Computer Programming	3	3/12 = 0.25	0.25*25 = 6	-	2	2	1	1	-	-	-	6
	Object Oriented Programming	3	3/12 = 0.25	0.25*25= 6	-	1	2	2	1	-	-	-	6
	Design and Analysis of Algorithms	3	3/12 = 0.25	0.25*25= 6	-	2	1	2	1	-	-	-	6
	Data Structure and Algorithms	3	3/12= 0.25	0.25*25= 7	-	2	1	2	2	-	-	-	7

[2] Fundamentals of Database Systems and Advanced Database Systems (3+3)

Themes	Course Name	Credit hour	Weight of course or proportion	Number of test items from each course	Cognitive						Affective	Psychomotor	Total
					Remember	Understand	Apply	Analyze	Evaluate	Create			
	Total credit	**12**											
Computer Networking and Security Share = 18 items from the total	Data Communication and Computer Networking	3	3/9 = 0.33	0.33*18 = 6	-	3	1	1	1	-	-	-	6
	Computer Security	3	3/9 = 0.33	0.33*18 = 6	-	2	3	1	-	-	-	-	6
	Network and System Administration	3	3/9 = 0.33	0.33*18= 6	-	1	1	2	1	1	-	-	6
	Total credit	**9**											
Intelligent Systems Share = 6 items from the total	Introduction to Artificial Intelligence	3	3/3=1	1*6=6	1	2	1	2	-	-	-	-	6
	Total credit	**3**											
Computer Architecture and Operating Systems Share = 12 items from the total	Operating System	3	3/6=0.5	0.5*12=6	-	1	2	1	2	-	-	-	6
	Computer organization and architecture	3	3/6=0.5	0.5*12=6	-	3	-	3	-	-	-	-	6
	Total Credit	**6**											
Compiler and Complexity Share = 12 items from the total	Automata and Complexity Theory	3	3/6=0.5	0.5*12=6	-	4	1	1	-	-	-	-	6
	Compiler Design	3	3/6=0.5	0.5*12=6	1	2	1	2	-	-	-	-	6
	Total credit	**6**											
Total for the program		49			1	31	23	23	15	7	-	-	100

Table 6-2*:* Mapping Learning Outcomes to Skill Sets

Themes	Course Name	General Objective	Learning outcomes	Cognitive						Affective	Psychomotor	Total
				Remember	Understand	Apply	Analyze	Evaluate	Create			
System Development Share = 27 items from the total	Software Engineering Number of test items from the course =6	Identifying a problem, studying the problem, gathering data and relevant materials and an open presentation of the development of the project work	Understand software analysis, design and development techniques and tools	-	2	-	-	-	-	-	-	2
			Demonstrate conceptual and technical skills in the analysis, design and implementation of a software system	-	-	1	1	1	-	-	-	3
			Create requirements using use case modeling concepts	-	-	-	-	-	1	-	-	1

Themes	Course Name	General Objective	Learning outcomes	**Cognitive**						**Affective**	**Psychomotor**	Total
				Remember	Understand	Apply	Analyze	Evaluate	Create			
	Web Programming Number of test items from the course =9	Develop the ability to plan, develop, test, and debug web pages logically.	Understand concepts, principles and methods in programming for web and Internet environment	-	2	-	-	-	-	-	-	2
			Apply Server Side scripting Languages and Implement client-side interactivity	-	-	2	-	-		-	-	2
			Specify, build and manage form and content of information-rich web sites	-	-	-	1		1	-	-	2
			Design, implement and evaluate client-server systems following specific protocol specifications, taking into account concurrency issue	-	-	-	1	2		-	-	3
	Fundamental of Database Systems Number of test items from the course =6	Introduce database management systems, with a focus on how to organize, maintain, and retrieve data efficiently and effectively	Understand the principles of database design	-	2	-	-	-	-	-	-	2
			Apply the database concepts to real world database design	-	-	1	1	-	-	-	-	2
			Design database systems for real world scenarios		-	-	-	-	2	-	-	2
	Advance Database Systems Number of test items from the course =6	Introduce the concepts of database system architecture, query optimization, parallel and distributed database systems.	Describe the main concepts of the object oriented model	-	1	-	-	-	-	-	-	1
			Use different recovery methods when there is a database failure	-	-	1	-	1	-	-	-	2
			Design a distributed database system in homogenous and heterogeneous environments	-	-	1	-	-	1	-	-	2
			Evaluate a set of query processing strategies	-	-	-	-	1	-	-	-	1

Themes	Course Name	General Objective	Learning outcomes	**Cognitive**						**Affective**	**Psychomotor**	Total
				Remember	Understand	Apply	Analyze	Evaluate	Create			
Programming and Algorithms Share = 25 items from the total	Computer Programming Number of test items from the course =6	Introduce a disciplined approach to problem solving methods and algorithm development, as well as to teach the syntax and vocabulary of a modern programming language	Understanding the principles of computer programming	-	2	-	-	-	-	-	-	2
			Design, code, debug and solve problems in using programming concepts	-	-	1	1	-	-	-	-	2
			Apply the principle of mathematical knowledge to prove statements and solve problems in computing science	-	-	1	-	1	-	-	-	2
	Object Oriented Programming Number of test items from the course =6	Demonstrate how real-world entities such as inheritance, hiding, polymorphism, and son can be implemented in programming	Understand the basic object oriented concepts	-	1	-	-	-	-	-	-	1
			Successfully code, debug and run programs with object oriented principles	-	-	2	2	-	-	-	-	4
			Apply object oriented programming concepts to solve problems	-	-	-	-	1	-	-	-	1
	Design and Analysis of Algorithms Number of test items from the course =6	Understanding of algorithm design and analysis for a variety of problems, as well as developing skills to reason about and prove algorithm properties such as correctness and running time.	Perform algorithm analysis using the different techniques	-	1	-	2		-	-	-	3
			Demonstrate the use of algorithm design techniques	-	-	1	-	1	-	-	-	2
			Describe the basics of computational complexity	-	1	-	-	-	-	-	-	1
	Data Structure and Algorithms Number of test items from the course =7 Total credit	Combine fundamental data structures and algorithmic techniques to create a complete algorithmic solution to a given problem.	Understand the most common concepts of data structures like stack, queue and linked list	-	2	-	1	-	-	-	-	3
			Implement the Sorting and searching concepts	-	-	1	1	1	-	-	-	3
			Use the concepts related to Data Structures and Algorithms to solve real world	-	-	-	-	1	-	-	-	1
Computer Networking and Security	Data Communication and Computer Networking	Understand the fundamentals of data communications networks, including protocol operation and the concepts and operation of local	Understand the concepts and principles of data communications and computer networks	-	1	-	-	-	-	-	-	1
			Understand Protocols and various networking components	-	1	-	-	-	-	-	-	1

Themes	Course Name	General Objective	Learning outcomes	**Cognitive**						**Affective**	**Psychomotor**	Total
				Remember	Understand	Apply	Analyze	Evaluate	Create			
Share = 18 items from the total	Number of test items from the course =6	and IP-based networks.	Understand TCP/IP & OSI Reference Model Understand	-	1	-	-	-	-	-	-	1
			Design and implement IP addressing and subnets	-	-	1	1	1	-	-	-	3
	Computer Security Number of test items from the course =6	Plan, implement, and monitor computer security mechanisms to help ensure the protection of information technology assets, as well as to protect and defend computer systems.	Understand the basic concepts in computer security	-	1	-	-	-	-	-	-	1
			Understand issues related to program security and the common vulnerabilities in computer programs	-	1	-	1	-	-	-	-	2
			Explain the basic requirements for trusted operating systems, and describe the independent evaluation, including evaluation criteria and evaluation process	-	-	-	1	-	-	-	-	1
			Describe security requirements for database security, and describe techniques for ensuring database reliability and integrity, secrecy, inference control, and multi-level databases.	-	-	-	1	1	-	-	-	2
	Network and System Administration Number of test items from the course =6	Prepare learners to manage the computer operations and control the system configurations emanating from a specific site or network hub.	Understand the concepts, principles, and roles of system and network administration	-	1	-	-	-	-	-	-	1
			Provide network services to users	-	-	1	1	-	-	-	-	2
			Identify security policies and troubleshooting	-	-	-	-	2	-	-	-	2
			Apply scripting for system administration	-	-	-	-	-	1	-	-	1
Intelligent Systems Share = 6 items from the total	Introduction to Artificial Intelligence Number of test items from the course =6	Describe the fundamental principles, techniques, and applications of artificial intelligence.	Understand reasoning, knowledge representation and learning techniques of artificial intelligence	1	1	-	-	-	-	-	-	2
			Assess the role of AI in gaining insight into intelligence and perception	-	-	1	-	-	-	-	-	1

Themes	Course Name	General Objective	Learning outcomes	**Cognitive**						**Affective**	**Psychomotor**	Total
				Remember	Understand	Apply	Analyze	Evaluate	Create			
			Understand the use of heuristics in search problems and games	-	1		1	-	-	-	-	2
			Evaluate the strengths and weaknesses of these techniques and their applicability to different tasks	-	-	-	1	-	-	-	-	1
Computer Architecture and Operating Systems Share = 12 items from the total	Operating System Number of test items from the course =6	Provide an overview of how modern operating systems work on the inside, focusing on processes and threads, mutual exclusion, CPU scheduling, deadlock, memory management, and file systems	Explain the objectives and functions of modern operating systems	-	1	-	-	-	-	-	-	1
			Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.	-	1	1	-	-	-	-	-	2
			Explain conditions that lead to deadlock	-	-	-	1	-	-	-	-	1
			Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes.	-	-	-	1	1	-	-	-	2
	Computer organization and architecture Number of test items from the course =6	Examine the structure and functions of the components comprising a contemporary computer system.	Identify different ways of communicating with I/O devices and standard I/O interfaces.	-	1	-	1	-	-	-	-	2
			Describe different performance enhancement of computer architecture	-	1	-	1	-	-	-	-	2
			Explain the basic structure of computer hardware & software	-	1	-	-	-	-	-	-	1
			Identify the processes involved in the basic operations of CPU	-	-	-	1	-	-	-	-	1
Compiler and Complexity Share = 12 items from the total	Automata and Complexity Theory Number of test items from the course =6	Understand methods for describing and analyzing the dynamic behavior of discrete systems, as well as concepts and methods for	Acquire insights into the relationship among formal languages, formal grammars, and automata.	-	1	-	-	-	-	-	-	1
			Identify different formal language classes and their relationships	-	1	-	-	-	-	-	-	1

Themes	Course Name	General Objective	Learning outcomes	Cognitive						Affective	Psychomotor	Total
				Remember	Understand	Apply	Analyze	Evaluate	Create			
		determining complexity in both time and space.	Design grammars and recognizer for different formal languages	-	1	1	1	-	-	-	-	3
			Understand Complexity classes, P/NP/PSPACE, reductions, hardness, completeness, hierarchy, relationships between complexity classes. complexity	-	1	-	-	-	-	-	-	1
	Compiler Design Number of test items from the course =6	Understand the fundamental principles of compiler design, its various constituent parts, algorithms, and data structures that must be used in the compiler, and provide the skills required for building compilers for various situations encountered in a computer science career.	Specify and analyze the lexical, syntactic and semantic structures of advanced language features	-	-	1	2	-	-	-	-	3
			Describe techniques for lexical and syntax analysis	1	-	-	-	-	-	-	-	1
			Understand the basic principles of compiler design	-	1	-	-	-	-	-	-	1
			Describe syntax directed translation and type checking techniques	-	1	-	-	-	-	-	-	1
Total for the program				1	31	23	23	15	7	-	-	100

7. Conclusion

Exit examination can play an important role in producing graduates who are knowledgeable, skilled, and emotionally mature. It helps to prepare competent graduates by serving as a quality check for effectiveness. It also contributes to the enhancement of the quality and effectiveness of academic programs. Furthermore, it can serve as a platform for collaboration among academic programs at various universities to work together to improve program quality.

To meet the graduation profile, competency and learning outcome, exit exam competency selection and identifying core course was done for 2015 EC graduating students. On top of that, a test blueprint demonstrating the share of themes, courses, and number of items for a total of 100 test items is presented. Furthermore, each course learning outcome is mapped with the skill sets to facilitate the preparation of fairly distributed items on each team, course, and learning outcome.

8. Appendix

8.1. Steps followed to develop test blueprint

i. The identified core competencies corresponding to the respective course were listed.
ii. Specific and minimum competencies to be assessed by the exam were identified.
iii. Major learned course contents corresponding to these listed minimum competencies were listed.
iv. Measurable learning outcomes based on these minimum competencies were identified using action verbs.
v. The test format was determined.
vi. The weighting of the listed learning outcomes was determined.
vii. Learning domain categories for which each of these learning outcomes was identified
viii. The test blueprint table was created.
ix. The number of test items for each learning outcome was determined by the types and levels of learning domains.
x. The sub-total and total test items were chosen based on their learning domains and learning outcome categories.

8.2. Issues considered in the test blueprint preparation

i. Curriculum details
ii. Course outlines/guidebook details, lists of learning outcomes and other curricular materials
iii. Course contents pertinent to the respective learning outcomes
iv. Time devoted to cover the content with in a course (course credit hour)
v. Total test length (i.e., total number of test items)
vi. Number of questions allocated to each content area using the proportion calculated
vii. Weight for each item
viii. Relevance of the content to students future career
ix. In addition to these, the documents listed below were considered:
    - Identified core competencies of the program
    - Harmonized curriculum of the program

8.3. Calculation of Share of the Themes/Courses/Items in Percentage

- Share of themes (T) = $\frac{a}{b}\ x100$, Where "a" is the credit hour of a theme and "b" is the total credit hour of the program. Credit hour of a theme is the sum of credit hours of courses in the theme.
    - Share of courses per theme (C) = $\frac{Credit\ hour\ of\ the\ course}{Credit\ hour\ of\ the\ theme}\ x100$
    - Share of items per course = $Share\ of\ the\ course\ X\ Total\ number\ of\ items$