



Tosa Skills Framework

Python

Overview

Introduction to Tosa Skills Framework	2
Tosa® (Test on Software Applications).....	3
Tosa Skills Framework Objective.....	3
Unique Tosa Scoring.....	3
Python Domains and subdomains.....	4
About the Python Certification	4
Tosa Sequence for Progressive Skills Development	6
Level 1 – Beginner User	7
Overview	8
Level 2 – Basic User	9
Language and syntax.....	10
Data structures and objects	10
Modules and packages.....	10
Code optimization	10
Overview	11
Level 3 – Productive User	12
Language and syntax.....	13
Data structures and objects	13
Modules and packages.....	13
Code optimization	13
Overview	14
Level 4 - Advanced User	15
Language and syntax.....	16
Data structures and objects	16
Modules and packages.....	16
Code optimization	16
Overview	17
Level 5 – Expert User	18
Language and syntax.....	19
Data structures and objects	19
Modules and packages.....	19
Code optimization	19
Overview	20

Introduction to Tosa Skills Framework

For Tosa Assessment and Certification

Tosa® (Test on Software Applications)

Tosa assessments and certifications will determine and validate a candidate’s proficiency and skill level in software development used in a professional environment. Tosa assessments are designed to validate the professional Python skills of individuals (students, trainees, employees, or jobseekers) in supporting their employment, professional, or academic objectives.

Tosa assessments employ the Adaptive Testing Methodology, which creates a personalized testing experience adapted to a candidate’s skill level for a selected software application. The score is based on the Item Response Theory using a 3-parameter logistic model, similar to the GMAT scoring method. Adaptive-based testing selects questions that challenge candidates to the limit of their knowledge and abilities.

Tosa Skills Framework Objective

This Tosa framework provides an overview of the subject areas being assessed during the Tosa Assessment and Certification exams. Tosa Python assessments and certification exams validate candidate proficiency in Python programming using a score on a scale from 1-1000 for the Certification Assessment, and a score divided into five levels, from “Beginner” to “Expert,” for the Diagnostic Assessment.

The objective of this document is to present an overview of the technical skills associated with each of the four main Python domains within each proficiency level. This information will also support educators and trainers in tailoring their training program to achieve desired proficiency levels.

Unique Tosa Scoring

The Tosa assessments and certifications are based on a unique score, divided into five levels.

- ranging from 1 to 1000 for the certification.
- divided into five levels, from Beginner to Expert, for assessment.

Tosa® levels	Corresponding Tosa® score	Certification status & documents issued
Expert	876 - 1000	Certification earned - diploma & Credly digital badge issued
Advanced	726 – 875	Certification earned - diploma & Credly digital badge issued
Productive	551 – 725	Certification earned - diploma & Credly digital badge issued
Basic	351 – 550	Certification failed - certificate of completion issued
Beginner	1 – 350	Certification failed - certificate of completion issued

Python Domains and subdomains

Language and syntax	<ul style="list-style-type: none"> ■ Context and use cases of Python ■ Syntax and semantics ■ Input/output management
Data structures and objects	<ul style="list-style-type: none"> ■ Import and use ■ Standard library
Modules and packages	<ul style="list-style-type: none"> ■ Classical structures ■ Functions and procedures ■ Object-oriented programming
Code optimization	<ul style="list-style-type: none"> ■ Code performance ■ Algorithms

About the Python Certification

The Tosa Python certification relies on a database of around 170 questions. It is composed of 35 questions and lasts one hour and a half. The algorithm adapts to each answer of the candidates to adjust the difficulty level of the questions until they reach the exact definition of the candidates' level by calculating the limit of their skills.

Since the test is adaptive, the series of questions that each candidate gets is unique for each test. This uniqueness allows for a more accurate evaluation of the candidate's level. It also limits cheating and the memorization of questions on different passages.

Our platform allows individuals to take the certification in class, in an approved testing center, or remotely via our integrated asynchronous online proctoring solutions.

Our remote proctoring solutions provide added flexibility for both the administrator and the candidate, allowing the certification exam to be taken anywhere, at any time. The candidate only needs an internet connection and a computer equipped with a working webcam and microphone.

Candidates receive a numeric score out of 1000 points associated to a proficiency level on a five-level scale. Candidates who score between 1 and 550 points don't earn the certification.

They will not receive a diploma but a certificate of completion. Candidates who score 551 points or above earn the certification. They will receive a diploma by email within five (5) business days and are eligible to a Credly digital badge.

There is no requirement to be eligible to take the exam, but our recommendations to be well prepared on exam day are:

- Take at least one Tosa Python adaptive assessment to estimate your level and get familiar with the test format
- Use free practice tests on our website for training
- Follow e-learning or training courses (average duration per level is between 10 and 15 hours per certification so around 150 hours total)

Tosa certification diplomas are valid for three years from the date of issue as skill levels evolve or decline over time, depending on the use of the software. New software and software versions are released every year, and skills must be updated. We cannot legitimately certify a digital skills level for more than three years. Limiting the certification validity reinforces the need for life-long learning and professional development.

Tosa certifications can be retaken when it is expired. Earners willing to improve their score and level can also retake the exam at any time.

Tosa Sequence for Progressive Skills Development













Students and professionals can tailor their certification journey with Tosa through vertical progression, demonstrating increasing proficiency in Python. Starting at a productive level, users can advance to advanced and expert levels as their skills grow. This clear path encourages continuous improvement and validates each stage of their development. Tosa's structure makes it easy to track progress and showcase evolving expertise.

Level 1 – Beginner User

Between 1 and 350 points

The Beginner Proficiency is set for a score from 1 to 350, which is the lowest Tosa score category. Achievement of Beginner score defines little or limited knowledge of Python, including the language’s basic functions and features, highlighting the inability of the candidate to use Python in a professional environment.

Overview

Domain	Skills Assessed
Language and syntax	<ul style="list-style-type: none">  Create a variable  Add a comment  Display the content of a variable  Use the range function  Use Loop Controls
Data structures and objects	<ul style="list-style-type: none">  Recognize simple built-in objects: strings, numbers  Recognize and create Booleans  Identify appropriate operations on collections
Modules and packages	<ul style="list-style-type: none">  Knowing the use of the packages  Import a package from its name
Code optimization	<ul style="list-style-type: none">  Return a solution in a standardized format  Traverse a List of Characters

Level 2 – Basic User

Between 351 and 550 points

Prior to the acquisition of the skills of the Basic level, the candidate will have mastered the skills of the Beginner level.

Language and syntax

Use flow control features to automate conditional and/or repetitive data processing.

Business application: Recognition of typical Python use cases. Implementation of data extraction process from a known and stable source.

Data structures and objects

Recognize and create simple built-in objects to create programs that manipulate data.

Business application: Automate data processing for which the input data is of a known and fixed basic type (e.g. numbers, strings)

Modules and packages

Master the concepts related to modules to include the functionality of a module in a program.



















Business application: Be able to create a program reusing elementary functions and objects created by another member of the company.

Code optimization

Store and process multiple data simultaneously to create a suitable simple data processing flow.

Business application: Without a previously created structure and based solely on a problem, be able to create a complete program that meets a set of specifications.

Overview

Domain	Skills Assessed
Language and syntax	<ul style="list-style-type: none">  Identify simple object types  Know the main use cases of Python  Use the arithmetic operators  Create a conditional flow control  Create simple loops  Use the global keyword
Data structures and objects	<ul style="list-style-type: none">  Recognize all structured built-in objects  Apply the min, max and len function  Create and manipulate a dictionary and a set  Understand Stack class methods
Modules and packages	<ul style="list-style-type: none">  Import a specific module from a package  Import and use functions from a module  Use these functions within a program
Code optimization	<ul style="list-style-type: none">  Read an input data  Read and pre-process multiple inputs  Choose the types of variables adapted to the problem  Combine different types of data  Analyze a depth first search algorithm

Level 3 – Productive User

Between 551 and 725 points

Prior to the acquisition of the skills of the Productive level, the candidate will have mastered the skills of the Basic level.

Language and syntax

Create a program for complete data processing, using the advanced features of flow controls and arithmetic operations.

Business application: Aggregation of available data from an existing extraction. Implementation of metrics relevant to the specific business need.

Data structures and objects

The candidates can recognize built-in structured objects (*list*, *set*, *dict*), to perform processing of multiple simple variables. They can manipulate simple, structured built-in objects and assemble them to create simple, reusable functions.

Business application: Create complex indicators from simple data (e.g. monthly monitoring of average income over a given period).

Modules and packages

At this level, candidates can select and import specific functions from package of the API for reuse in data processing. They also have the ability of creating a complete simple package to create shareable and reusable features.

Business application: integrate a specific tool from the standard library to solve a classical problem (date processing, simple mathematical functions for example).

Code optimization

Model a problem and automate data manipulations, in order to process a large or complex volume of data.

Business application: Mobilize basic algorithmic concepts to accelerate the processing of large volumes of data, which would be too slow to be useful in business if solved naively.

Overview

Domain	Skills Assessed
Language and syntax	<ul style="list-style-type: none"> 🔧 Know the different versions of Python 🔧 Master the order of operations 🔧 Create the necessary variables 🔧 Master the flow control instructions associated with loops 🔧 Read data from the standard input
Data structures and objects	<ul style="list-style-type: none"> 🔧 Document a function 🔧 Perform an iteration on a structured object 🔧 Handle functions associated with structured objects 🔧 Create functions 🔧 Use Decorator Functions
Modules and packages	<ul style="list-style-type: none"> 🔧 Know and use the math and random packages 🔧 Create a module 🔧 Distinguish between internal components and modules 🔧 Distinguish the main packages of the standard library
Code optimization	<ul style="list-style-type: none"> 🔧 Sort data 🔧 Model a graph and explore it 🔧 Create combinations of loops and conditions for different problems 🔧 Automatically extract information from a large volume of data 🔧 Identify the commonly used tool for static code analysis 🔧 Write a Cipher Encoder

Level 4 - Advanced User

Between 726 and 875 points

Prior to the acquisition of the skills of the Advanced level, the candidate will have mastered the skills of the Productive level.

Language and syntax

The Advanced-Users candidates know the syntactic and semantic rules of Python to create clear and reusable programs. Master the input-output tools, to read and modify documents external to the program.

Business application: Automation of data processing from external documents and saving this processing for archiving or future reuse.

Data structures and objects

The candidates master the specifics of integrated types to process large volumes of data efficiently and clearly. They create documented functions, classes, and their associated methods, to create functionalities that can be integrated into other programs.

Business application: Create specific objects, to lay the technical foundations of a complex project, and make its bases as maintainable and shareable as possible.

Modules and packages

At this level, the candidates can import multiple complex packages to exploit all the included features. They can use the basic packages of the standard library to efficiently manipulate common external data.

Business application: use several tools from the same package of the standard library to solve a complex problem. For example: generate specific random data with random or solve expressions numerically with math.

Code optimization

Implement suitable data structures and choose suitable functions to design programs with controlled algorithmic complexity.

Business application: Know how to create specific data structures to respond as efficiently as possible to problems involving large volumes of complex data. Know how to distinguish the algorithmic complexity of several existing programs (constant, linear, quadratic) to select the most efficient option.

Overview

Domain	Skills Assessed
Language and syntax	<ul style="list-style-type: none"> ✎ Distinguish the core features of the Python language ✎ Master literals ✎ Know the syntactic rules on variables ✎ Use all printing features on standard and error output ✎ Read and modify a file
Data structures and objects	<ul style="list-style-type: none"> ✎ Define structures by comprehension ✎ Choose the right type of variable according to mutability ✎ Create f-strings on different types ✎ Design functions with different types of arguments ✎ Create a class and its initialization
Modules and packages	<ul style="list-style-type: none"> ✎ Navigate in a package with the dir function ✎ Install new packages with pip ✎ Distinguish use cases from most of the standard library packages ✎ Reuse data structures and functions from the standard library
Code optimization	<ul style="list-style-type: none"> ✎ Create optimized data structures for different problems ✎ Use the most efficient standard functions as appropriate ✎ Distinguish the algorithmic complexity of simple programs (constant, linear, quadratic)

Level 5 – Expert User

Between 876 and 1000 points

Language and syntax

The Expert-Users can integrate error handling to create programs that cover all cases of data entry and data processing. They can use naming rules and conventions to create programs that can be integrated into a complete environment.

Business application: Refine an existing program to handle errors and unknown data in a readable way.

At this level, the candidate can train on the language.

Data structures and objects

At this level, the candidates implement the full range of functional tools, to create adapted and efficient functions. They can use and implement the concepts of object-oriented programming to create project-specific structures.

Business application: Automate the most complex tasks of a project, while maintaining a high level of maintainability; create complex objects for specific business applications that can be perfectly integrated into a large-scale project.

At this level, the candidate can train on the language.

Modules and packages

The candidates master all the concepts related to modules to integrate packages in the most efficient way possible. They use the entire standard library to process external data and documents, locally or on the Internet.

Business application: Create external data processing of various formats (CSV or JSON in particular), by browsing all the local files, or on the internet; be able to operate and maintain technical components, even when they are poorly documented.

At this level, the candidate can train on the language.



















Code optimization

The candidates can implement advanced algorithmic tools to design programs that use the least amount of resources possible.

Business application: Implement advanced algorithmic tools to drastically improve the performance of an existing program. At this level, the candidate can train on the software.

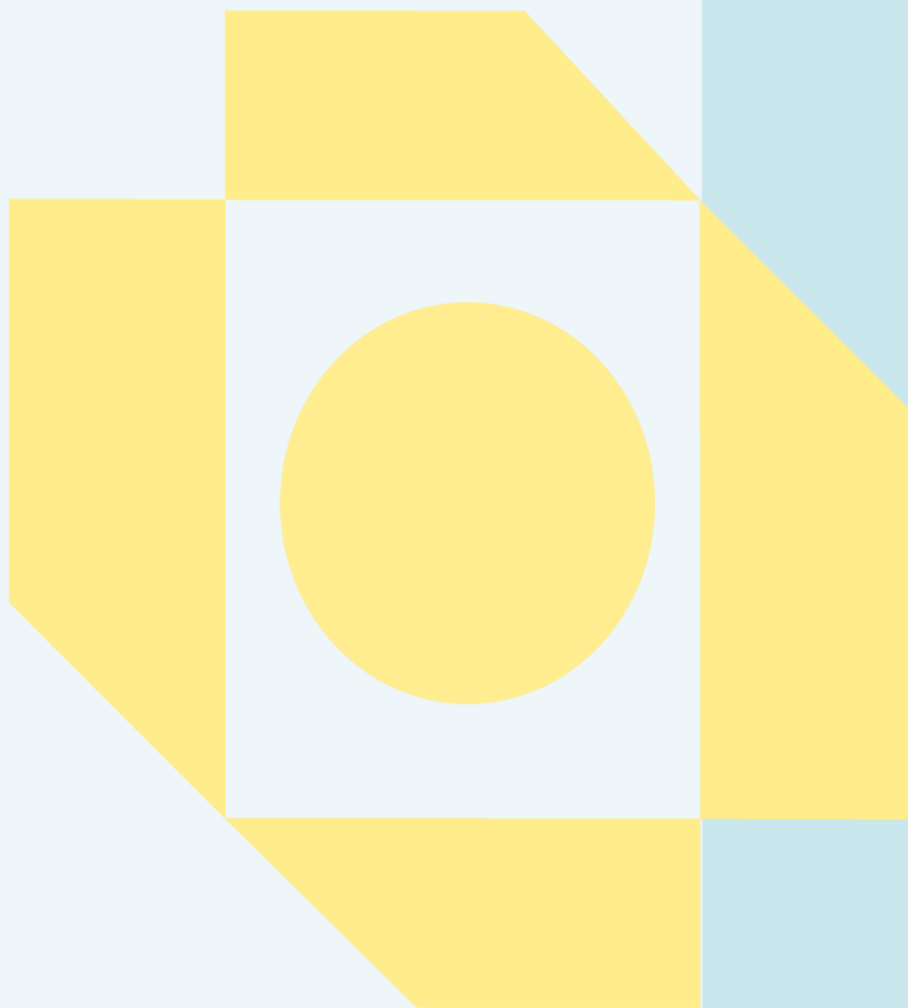
At this level, the candidate can train on the language.

Overview

Domain	Skills Assessed
Language and syntax	<ul style="list-style-type: none">  Capture and manage errors  Distinguish and create different types of errors  Integrate errors into flow controls  Differentiate between the priorities of scopes and namespaces  Distinguish Python environments
Data structures and objects	<ul style="list-style-type: none">  Use lambda functions  Use decorators  Create and use generators  Define class-specific methods  Create a data structure adapted to a problem  Manage inheritance between different classes
Modules and packages	<ul style="list-style-type: none">  Recognize the compiled files and their interest  Import functions with internal references  Interact with the internet via a script  Process documents in different formats  Use the sys and os packages
Code optimization	<ul style="list-style-type: none">  Analyze and optimize existing code  Recognize an application case and use an appropriate data structure



Your skills. Your advantage.



contact@isograd.com

www.tosa.org