

UDC 004.43:378(476-25)

COMPARATIVE ANALYSIS OF PROGRAMMING LANGUAGES STUDIED AT BSUIR AND THEIR COMPLIANCE WITH MODERN REQUIREMENTS OF THE IT MARKET

Holod I.V., Zhukovskaya P.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Lasarenko A.M. – Senior Lecturer at the Department of Foreign Languages

Annotation. The gap between what universities teach and what IT employers actually need is the central concern of this paper. Using BSUIR as a case study, the authors examine official faculty syllabi to determine which programming languages are covered and how relevant each remains in 2025. Graduate preparedness is then assessed in the context of industry shifts driven by artificial intelligence and high-performance systems. Popularity indices (TIOBE, PYPL, IEEE Spectrum 2024–2025) and job-listing data from Belarusian recruitment platforms provide the empirical basis for a scored comparison between the academic programme and current employer expectations.

Key words: IT education; BSUIR; technology stack; IT labour market 2025; programming languages; industrial development; curriculum alignment.

Introduction. The accelerating pace of technological change has placed universities in an uncomfortable position: curricula that take years to revise must somehow remain meaningful in a field where dominant tools can shift within months. At BSUIR, this challenge has traditionally been addressed by building the programme on a foundation of classical computer-science theory while progressively supplementing it with courses tied to real-world software production.

The Stack Overflow Developer Survey 2024 confirms that hiring teams prefer candidates with hands-on command of tools already running in production. By the time a new technology enters a university syllabus, it may already be widely adopted in the workplace, creating a measurable lag between academic programmes and industry practice [1].

Rather than a general critique of IT education, this study delivers a granular, data-supported audit of specific programming languages: which ones are taught, how popular they are in the current job market, and where the syllabus has fallen behind what employers now require.

The core aim is to produce a structured comparison between the language content of BSUIR's programme and the technology preferences documented in employer hiring data, thereby determining how well-positioned graduates are when they enter the labour market.

Three tasks support this aim: a content audit of language-related modules across the key BSUIR faculties; a cross-tabulation of the resulting academic technology stack with leading industry ranking systems; and the identification of curriculum gaps accompanied by specific remediation proposals.

The authors proceed from the hypothesis that BSUIR's foundational language instruction is adequate, but that the programme has not kept pace with the employer-driven surge in demand for TypeScript, Rust, and cloud-native infrastructure skills, creating a gap that lengthens graduates' time-to-productivity after hiring.

Main part. Three evidence streams are combined: (a) official BSUIR teaching documentation, (b) numerical data from job-listing portals and international language ranking indices, and (c) a criterion-weighted scoring model applied to each language in the curriculum.

Primary data were drawn from the following sources:

- official syllabi for the BSUIR degree programmes «Software Engineering» and «Artificial Intelligence», academic year 2024–2025;
- three widely cited language-popularity measurement systems: TIOBE Index, PYPL Index, and the IEEE Spectrum Top Languages ranking (2024–2025 editions);
- job-posting datasets extracted from the dominant IT recruitment platforms operating in Belarus and neighbouring CIS countries (dev.by, rabota.by, hh.by);

– the Stack Overflow Developer Survey 2024, specifically its sections on languages that respondents most value, most avoid, and most wish to learn.

Analytical framework. Each language present in the BSUIR curriculum was assigned a conformity coefficient C , calculated according to the formula:

$$C(\text{lang}) = w_1 \cdot P(\text{lang}) + w_2 \cdot V(\text{lang}) + w_3 \cdot T(\text{lang}), \quad (1)$$

where $P(\text{lang})$ – normalised composite popularity score derived from TIOBE and PYPL; $V(\text{lang})$ – share of regional job postings that list the language as a requirement; $T(\text{lang})$ – directional trend indicator (upward growth = 1, flat = 0.5, declining = 0); w_1, w_2, w_3 – weights set empirically at 0.35, 0.40, and 0.25 respectively.

Languages were then grouped into three tiers: High Market Fit ($C > 0.7$), Partial Market Fit ($0.4 \leq C \leq 0.7$), and Low Market Fit ($C < 0.4$).

Results: Architecture of the Educational Stack. Foundational tier – C/C++. The university's choice to keep C and C++ as first-year anchors stands up to scrutiny against 2025 market data. C++ currently occupies position 10 in TIOBE, yet the metric understates its renewed relevance: both Edge Computing expansion and growing demand to optimise LLM inference pipelines are driving fresh uptake of manually managed, low-overhead C++ code in performance-sensitive systems.

C++ also carries pedagogical value: hardware architecture, pointer arithmetic, and complexity analysis in one sequence – competencies transferable across all specialisations. For embedded or systems-level roles: $C(C++) \approx 0.78$ (High Market Fit).

Industrial standard – Java and C#. Java and C# (.NET) enter from the second year, oriented toward large-scale enterprise development.

In fintech and enterprise backend engineering, Java retains unquestioned dominance. Senior-year Spring Framework coursework closes most of the gap between academic training and production requirements, translating into an estimated 85–90 % readiness for a Junior+ Java role. Regional job boards confirm its standing: 31 % of all back-end vacancies reference Java explicitly [2]. Score: $C(\text{Java}) \approx 0.82$ (High Market Fit).

BSUIR's Unity and enterprise .NET tradition ensures C# graduates are employable in game studios and Azure projects. $C(C\#) \approx 0.71$ (High Market Fit, lower boundary).

Future vector – Python. On the FCSN and FITC faculties, Python is the designated language for Big Data coursework and neural-network programming. This positioning mirrors the broader industry consensus: Python holds the top spot in both TIOBE and PYPL for 2024–2025 and commands the machine learning, data science, and automation segments with no serious rival in sight.

$C(\text{Python}) \approx 0.91$ – highest in the BSUIR stack. Advanced modules covering NumPy, Pandas, and TensorFlow give graduates a production-ready toolkit for data-intensive roles.

Table 1 shows scores and market indicators for each language.

Table 1 – Market conformity scores for programming languages in the BSUIR curriculum (2025)

Language	TIOBE Rank (2025)	Regional Vacancies	Trend	$C(\text{lang})$	Tier
C / C++	1 / 10	18 %	↑	0.78	High
Java	4	31 %	→	0.82	High
C#	7	22 %	→	0.71	High
Python	1 (PYPL)	39 %	↑	0.91	High
JavaScript	7	27 %	→	0.68	Moderate
SQL	–	45 %	→	0.75	High
Rust	13	5 %	↑↑	0.44	Moderate*
TypeScript	–	21 %	↑	0.38	Low*

* – included in the BSUIR curriculum as optional or introductory-level electives only.

Discussion: Critical Analysis and Blind Spots. While the programme's foundations are strong, three specific areas emerge from the data as systematic under-investments that slow graduate onboarding in the fastest-growing segments of the IT market.

Frontend and TypeScript deficit. TypeScript– the statically typed superset of JavaScript now dominant in large-scale frontend and full-stack codebases– has been relegated to optional modules or brief mentions within broader web courses– a peripheral position that contradicts market realities.

Employer data shows the language appearing in 21 % of regional vacancies and ranking fifth globally by active usage in the Stack Overflow 2024 survey [3].

The resulting score, $C(\text{TypeScript}) \approx 0.38$, places it in the Low Market Fit category and marks the widest single gap between BSUIR's curriculum and current employer expectations.

Senior-level IT roles now routinely demand fluency in cloud platforms and container orchestration.

A survey of Belarusian vacancy postings shows that AWS, Google Cloud, or Azure combined with Docker, Kubernetes, and Terraform pipelines appear as requirements in 34 % of senior developer listings [2].

BSUIR's core curriculum makes no provision for any of these tools in mandatory subjects; interested students must seek them out in optional workshops, a situation that leaves cloud and DevOps aspirants significantly underprepared relative to market expectations.

Rust has topped the Stack Overflow most-admired language list for nine consecutive years through 2024, with growing adoption in Linux kernel modules and cloud-infrastructure components [4].

Since BSUIR already teaches C/C++ extensively, introducing a mandatory Rust module in third or fourth year would require minimal additional preparation while noticeably strengthening graduates' competitiveness in systems engineering roles.

Conclusion. BSUIR's core language offering aligns well with current IT labour market demands. Graduates leave with a T-shaped profile: deep expertise in Java or C++ combined with working knowledge of Python, SQL, and JavaScript [5].

Taken together, the findings confirm the initial hypothesis.

Core language instruction is solid, but three specific gaps– TypeScript, Rust, and cloud-native tooling– hold the mean conformity coefficient down to $\bar{C} \approx 0.68$, a score that places the full curriculum at the top of the Partial Market Fit zone, just short of the threshold for a High Market Fit rating.

Four actions are recommended: (1) upgrade TypeScript to a required subject in web and full-stack tracks; (2) embed a Rust module as a mandatory component of systems-programming specialisations; (3) introduce a core Cloud Computing course covering AWS or Azure fundamentals and Docker/Kubernetes; (4) establish a standing industry advisory channel so curriculum updates are triggered by labour-market shifts rather than administrative schedules.

The methodology applied here is transferable to other technical universities in Belarus and the CIS region seeking to audit their own curricula.

References

1. Stack Overflow. Developer Survey 2024. [Electronic resource]. – Mode of access: <https://survey.stackoverflow.co/2024>. – Date of access: 21.02.2026.
2. Dev.by Analytics. IT Vacancy Market in Belarus 2024–2025. – Minsk, 2025, pp.10-25.
3. TIOBE Software. TIOBE Programming Community Index. [Electronic resource]. – Mode of access: <https://www.tiobe.com/tiobe-index>. – Date of access: 21.02.2026.
4. IEEE Spectrum. Top Programming Languages 2024. [Electronic resource]. – Mode of access: <https://spectrum.ieee.org/top-programming-languages>. – Date of access: 21.02.2026.
5. Ivanov I.I. Methods of Teaching Object-Oriented Programming Languages in Technical Universities. – Minsk: BSUIR, 2023, pp.45-70.