

1. Kurso apžvalga, C++ standartai, kas yra objektiškai orientuotas programavimas (OOP)?, objekto koncepcija, pažintis su programavimo aplinka: kompiliatorių ir įrankių apžvalga, versijų kontrolės sistemos (git), make/cmake įrankiai, programų Setup projektų kūrimas.	2				1		3	4	Literatūros analizė, laboratoriniai darbai, interaktyvios medžiagos nagrinėjimas, praktikavimas sprendžiant užduotis: https://emokymai.vu.lt https://leetcode.com/ www.topcoder.com https://projecteuler.net/problems
2. Baziniai duomenų tipai, tipų transformacija, l-reikšmės ir r-reikšmės, rodyklės ir nuorodos, konstantinės nuorodos, dinaminis atminties valdymas, aritmetika su adresais, "išmaniosios" rodyklės, funkcijų persidengimas, direktyvos, įvesties ir išvesties operatoriai.	4				2		6	7	
3. Vartotojo tipai, klasės ir objektai, struktūros, konstruktoriai, pagrindinis konstruktorius, destruktoriai, objektinis projektavimas, „RAII“ paradigma, inkapsuliavimas, matomumo kontrolė, UML diagramos, dokumentacijos kūrimas su Doxygen.	4				2		6	10	
4. Operatorių persidengimas, įvesties/išvesties operatoriai, operatorių persidengimo realizavimo strategijos, kopijavimo konstruktorius, priskyrimo ir kopijavimo konstruktoriaus palyginimas, sekus ir gilus kopijavimas, nuorodos į r-reikšmes, objektų perkėlimo semantika.	4				2		6	6	
5. Kompozicija ir agregavimas, paveldėjimas, paveldėjimo kontrolė, konstruktoriai ir paveldėjimas, polimorfizmas, virtualiosios funkcijos, ankstyvas ir vėlyvas saistymas.	4				2		6	6	
6. Standartiniai išvesties ir įvesties srautai, failų srautai.	2				1		3	1	
7. Klaidų ir išimčių valdymas, laiko matavimas (<i>std::chrono</i> biblioteka), programos spartos matavimo įrankiai, atsitiktinių skaičių generavimas.	4				2		6	6	
8. Objektiškai orientuotas dizainas, interfeisai, bendrinis programavimas, šablono klasės ir funkcijos.	4				2		6	6	
9. Konteineriai (<i>vektorius, dekas, sąrašas, žemėlapiai, maišos lentelės</i> ir t.t.), <i>iteratoriai</i> . Algoritmų apžvalga: efektyvumas su algoritmais, sparta su duomenų struktūromis. Išimtinių situacijų apdorojimas, laiko matavimas, atsitiktinių skaičių generatoriai.	4				2		6	8	
10. Pasiruošimas egzaminui.								23	Literatūros kartojimas
Iš viso	32				16		48	77	

Vertinimo strategija	Svoris proc.	Atsiskaitymo laikas	Vertinimo kriterijai
Pirmasis laboratorinis darbas	30	Semestro metu	Studentams skiriamos individualios užduotys, apimančios 1-3 temas. Maksimalus įvertinimas už puikiai atliktas užduotis yra 10 balų (atitinkantys 30 % bendrojo svorio). Skiriami papildomi balai (iki 20 % maksimalaus įverčio svorio), jei užduotys atsiskaitomos anksčiau nurodyto termino. Analogiškai, vėluojant atsiskaityti galutinis įvertinimas yra mažinamas (iki 20 % maksimalaus įverčio svorio).
Antrasis laboratorinis darbas	20	Semestro metu	Studentams skiriamos individualios užduotys, apimančios 4-6 temas. Maksimalus įvertinimas už puikiai atliktas užduotis yra 10 balų (atitinkantys 20% bendrojo svorio). Skiriami papildomi balai (iki 20 % maksimalaus įverčio svorio), jei užduotys atsiskaitomos anksčiau nurodyto termino. Analogiškai, vėluojant atsiskaityti galutinis įvertinimas yra mažinamas (iki 20 % maksimalaus įverčio svorio).
Trečiasis laboratorinis darbas	10	Semestro metu	Studentams skiriamos individualios užduotys, apimančios 7-9 temas. Maksimalus įvertinimas už puikiai atliktas užduotis yra 10 balų (atitinkantys 10% bendrojo svorio). Skiriami papildomi balai (iki 20 % maksimalaus įverčio svorio), jei užduotys atsiskaitomos anksčiau nurodyto termino. Analogiškai, vėluojant atsiskaityti galutinis įvertinimas yra mažinamas (iki 20 % maksimalaus įverčio svorio).
Egzaminas (raštu)	40	Egzaminų sesijos metu	Egzaminą laikyti leidžiama semestro metu surinkus ne mažiau 7.5 balų skaičių, atitinkantį 25% laboratoriniams darbams skirtą svorį. Egzamino metu galima surinkti iki 10 taškų, kurie atitinka 40% galutinio įvertinimo. Egzaminas susideda iš dviejų etapų. Pirmiausia, studentas turi atsakyti į klausimus iš paskaitose pateiktų temų (iki 2 taškų). Antroje egzamino dalyje studentas turi pateikti praktinį pateiktos problemos sprendimą C++ kalboje (iki 8 taškų), motyvuojant naudojamų priemonių efektyvumą bei analizuojant alternatyvius užduoties sprendimo būdus.

Autorius	Leidimo metai	Pavadinimas	Periodinio leidinio Nr. ar leidinio tomas	Leidimo vieta ir leidykla ar internetinė nuoroda
Privaloma literatūra				
Bjarne Stroustrup	2014	Programming: Principles and Practice Using C++	2nd Edition	Addison-Wesley, http://www.stroustrup.com/programming.html
Stanley Lippman, Joséé Lajoie, and Barbara E. Moo	2012	C++ Primer	5th Edition	Addison-Wesley, http://www.informit.com/store/c-plus-plus-primer-9780321714114
Bjarne Stroustrup	2013	The C++ Programming Language	4th Edition	Addison-Wesley, http://www.stroustrup.com/4th.html
Bjarne Stroustrup	2022	Tour of C++, A (C++ In-Depth Series)	3rd Edition	Addison-Wesley, https://www.stroustrup.com/tour3.html
Papildoma literatūra				
Nicolai Josuttis	2012	C++ Standard Library Tutorial and Reference	2nd Edition	Addison Wesley Longman, http://cppstdlib.com/
Anthony Williams	2012	C++ Concurrency in Action: Practical Multithreading		Manning Publications Co
Scott Meyers	2014	Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14		O'Reilly, http://shop.oreilly.com/product/0636920033707.do
Scott Meyers	2005	Effective C++: 55 Specific Ways to Improve Your Programs and Designs	3rd Edition	Addison-Wesley, http://www.aristeia.com/books.html
Scott Meyers	2001	Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library		Addison-Wesley, http://www.aristeia.com/books.html
Herb Sutter	1999	Exceptional C++: 47 Engineering Puzzles, Programming Problems, and Solutions		http://www.gotw.ca/publications/xc++.htm
Andrew Koenig and Barbara Moo	2000	Accelerated C++: Practical Programming by Example		Addison-Wesley, http://www.informit.com/store/accelerated-c-plus-plus-practical-programming-by-example-9780201703535

1. Course overview, C++ standards, what is object-oriented programming (OOP), concept of the object, choosing the right IDE and compiler, version control systems (git), make/cmake tools, unit-testing, development of Application Setup projects.	2				1		3	4	Analysis of the literature, laboratory works, exploring of an interactive content, practice coding and problem-solving: https://emokymai.vu.lt https://leetcode.com/ www.topcoder.com https://projecteuler.net/problems
2. Core language types and type conversions, l-values and r-values, pointers, dynamic allocation, address arithmetic, array pointers, (constant) references, smart pointers, overloaded functions, inline functions, returning by reference, directives, output and input operators, namespaces.	4				2		6	7	
3. User-defined types, classes, struct, objects, constructors, default constructor, destructors, object-oriented design, C++ garbage collector through RAII paradigm, encapsulation, access specifiers, this pointer, static member variables and static member functions, UML diagrams, generating documentation with Doxygen.	4				2		6	10	
4. Operator overloading, overloading the I/O operators, overloading operators using standard and member functions, copy constructor, assignment vs. copy constructor, shallow copy vs. deep copy, r-value references, move semantics, "rule of 3" and "rule of 5"	4				2		6	6	
5. Object composition and aggregation, class inheritance, inheritance and access specifiers, constructors and initialization of derived classes, polymorphism, virtual functions, early and late binding.	4				2		6	6	
6. Standard output and input streams, std::string, file streams.	2				1		3	1	
7. Error and exception handling, measuring performance (std::chrono library), profiling tools, random number generator(s).	4				2		6	6	
8. Object-oriented design and patterns, interface, generic programming, template classes and functions.	4				2		6	6	
9. An overview of containers (vector, deque, set, map, hashables and etc.), iterators, and algorithms. Efficiency with algorithms, performance with data structures.	4				2		6	8	
10. Preparation for the exam and taking the exam.								23	
Total	32				16		48	77	

Assessment strategy	Weight ,%	Deadline	Assessment criteria
The first laboratory work	30	During the semester	Individual works are assigned to students covering topics 1-3. The maximum score for the assignment is 10 points (this corresponds 30% of the total weight). Students can receive bonus points (up to 20% of the maximum score) when tasks are successfully defended before the deadline. Similarly, the final assessment can be reduced (up to 20% of the maximum score) due to delays.
The second laboratory work	20	During the semester	Individual works are assigned to students covering topics 4-6. The maximum score for the assignment is 10 points (this corresponds 20% of the total weight). Students can receive bonus points (up to 20% of the maximum score) when tasks are successfully defended before the deadline. Similarly, the final assessment can be reduced (up to 20% of the maximum score) due to delays.
The third laboratory work	10	During the semester	Individual works are assigned to students covering topics 7-9. The maximum score for the assignment is 10 points (this corresponds 10% of the total weight). Students can receive bonus points (up to 20% of the maximum score) when tasks are successfully defended before the deadline. Similarly, the final assessment can be reduced (up to 20% of the maximum score) due to delays.
Written examination	40	During the exams session	The final exam is allowed to take if the minimum qualifying mark (equal to 7.5 points; equivalently to 25% of the total score from laboratory works). During the exam, students can collect up to 10 points, which corresponds to 40% of the final assessment. The examination consists of two stages. First, the student must answer different complexity questions corresponding topics covered on lectures (student can earn up to 2 points). In the second part, the student must provide a practical solution to the

			actual problem (real-world situation) (student can earn up to 8 points). Performance analysis, as well as analysis of alternative solution strategies, must be provided.
--	--	--	--

Author	Year of publication	Title	Issue of a periodical or volume of a publication	Publishing place and house or web link
Compulsory reading				
Bjarne Stroustrup	2014	Programming: Principles and Practice Using C++	2nd Edition	Addison-Wesley, http://www.stroustrup.com/programming.html
Stanley Lippman, Josée Lajoie, and Barbara E. Moo	2012	C++ Primer	5th Edition	Addison-Wesley, http://www.informit.com/store/c-plus-plus-primer-9780321714114
Bjarne Stroustrup	2013	The C++ Programming Language	4th Edition	Addison-Wesley, http://www.stroustrup.com/4th.html
Bjarne Stroustrup	2022	Tour of C++, A (C++ In-Depth Series)	3rd Edition	Addison-Wesley, https://www.stroustrup.com/tour3.html
Optional reading				
Nicolai Josuttis	2012	C++ Standard Library Tutorial and Reference	2nd Edition	Addison Wesley Longman, http://cppstdlib.com/
Anthony Williams	2012	C++ Concurrency in Action: Practical Multithreading		Manning Publications Co
Scott Meyers	2014	Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14		O'Reilly, http://shop.oreilly.com/product/0636920033707.do
Scott Meyers	2005	Effective C++: 55 Specific Ways to Improve Your Programs and Designs	3rd Edition	Addison-Wesley, http://www.aristeia.com/books.html
Scott Meyers	2001	Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library		Addison-Wesley, http://www.aristeia.com/books.html
Herb Sutter	1999	Exceptional C++: 47 Engineering Puzzles, Programming Problems, and Solutions		http://www.gotw.ca/publications/x_c++.htm
Andrew Koenig and Barbara Moo	2000	Accelerated C++: Practical Programming by Example		Addison-Wesley, http://www.informit.com/store/accelerated-c-plus-plus-practical-programming-by-example-9780201703535