

# SYLLABUS OF THE ACADEMIC DISCIPLINE "COMPUTER SCIENCE"

<b>Academic degree</b>	<b>Bachelor</b>
<b>Academic program</b>	<b>Electric Power Engineering, Electrical Engineering and Electromechanics</b>
<b>Duration</b>	<u>I-III quarter</u>
<b>Classes:</b>	<u>2022 - 2023 years of study</u>



## Type of classes:

**Lecture:** 2 hours a week - I-III quarter

**Practical:** 1 hours a week - I-II quarter


**Laboratory:** 2 hours a week - I-II quarter  
1 hours a week - III quarter

<b>Language</b>	<u>English</u>
<b>Department</b>	<u>Information technology and computer engineering</u>

## Distance courses

<https://do.nmu.org.ua/course/view.php?id=3446>

## Information about lecturers:

	<b>Kashtan Vita Yuriivna</b>	Associate Professor Candidate of Technical Sciences
	<b>Personal page</b>	<a href="https://it.nmu.org.ua/ua/HR_staff/prepods/kashtan.php">https://it.nmu.org.ua/ua/HR_staff/prepods/kashtan.php</a>
	<b>E-mail:</b>	Kashtan.V.Yu@nmu.one

## **1. Course abstract**

In recent years, powerful tools for verifying software and hardware systems have been developed. Computer Engineering has been contributing the world for the overall development and creating job or employment opportunities in both public and private sectors. This curriculum is designed to foster knowledge and skills to the technician required by the compute engineering and information technology related industries, electrical engineering.

This course covers the fundamentals of computer and mobile device hardware and software, and advanced concepts such as security, networking, programming and the responsibilities of an IT professional. Students who complete this course will be able to describe the internal components of a computer, assemble a computer system, install operating systems, and troubleshoot them using software tools and diagnostics. Students will also be able to connect to the Internet and share resources in a networked environment. New topics in this version include scripting basics, using remote access technologies, IoT device configuration and communication types, documentation and change management best practices, and also, disaster prevention and recovery methods.

Upon completion of the “Computer engineering and programming” course, students will be able to perform the following tasks:

- choose appropriate computer components for creating, repairing, or upgrading personal computers;
- install and configure components for assembling, repairing, or upgrading personal computers;
- to search for malfunctions on personal computers;
- explain how to troubleshoot laptops and other mobile devices;
- install the printer according to the requirements;
- describe virtualization and cloud computing;
- install Windows operating systems;
- manage and maintain Windows operating systems;
- explain how to configure, protect and troubleshoot mobile and operating systems;
- development of simple console programs based on acquired knowledge of building algorithms and programming skills in C++, and Visual Basic;
- development of programs with a graphical user interface in the Visual Basic language.

## **2. The purpose and objectives of the discipline**

The purpose of the discipline " Computer engineering and programming " is introduce to the computer hardware and software, operating systems, and programming elements to solve the problems of increasing the efficiency of the use of electric power, electrotechnical, and electromechanical equipment

Course objectives:

- to acquaint students with higher education with the principles of construction and operation of computer equipment;
- learn the simple components of a personal computer;
- teach students to configure PCs: install components for creating, repairing, or upgrading personal computers; choose appropriate computer components for creating, repairing, or upgrading a personal computer;
- teach how to troubleshoot personal computers;
- to learn to develop algorithms for their solution based on a verbal description of problems;
- to acquaint students with the basics of building system software;
- a study of technologies, means, and methods of programming in C++ and Visual Basic languages;
- to teach students to apply technologies for solving application problems in areas of professional activity.

### **3. Learning outcomes**

1. The ability to solve general and complex specialized tasks and practical problems for effective work with modern computer tools.
2. Ability to integrate knowledge, and solve complex issues, formulate judgments regarding insufficient or limited information.
3. Know the architecture and principles of operation of personal computers.
4. Know the stages of performing tasks on a PC, and the concepts of algorithmization and programming in C++ and Visual Basic.
5. To be able to clearly and unambiguously reveal conclusions and knowledge, arguing them, to professional and non-professional audiences.
6. To carry out effective communication and cooperation with colleagues, end users, management, specialists, and experts of various levels of other fields of knowledge, to express their opinions in a reasoned manner regarding current or future tasks in the professional field
7. Justify the choice of software products, their interaction, and potential impact on solving organizational problems, and implement their implementation and use in compliance with professional and ethical standards.

### **4. The structure of the discipline**

#### **LECTURES**

<i>1 semester</i>	
<b>LECTURES</b>	
Lecture 1	<b>1. Introduction to computer engineering and programming</b>
	Information, its types and properties.
	The concept of personal computers, their role in computer science.

	Classification of computer architectures. Von Neumann's architecture
	The main areas of application of computer technology in the field of electrical engineering.
	Significance and main directions of application of computer technology in the field of electrical engineering.
Lecture 2	<b>2. Personal Computer Hardware</b>
	Case and power supplies
	Electricity: Ohm's law. Voltage fluctuations in electrical networks.
	Motherboard Components
	CPUs and Cooling Systems
	Types of Memory
	Adapter Cards and Expansion Slots
	Hard disk drives and SSDs
	Optical Storage Devices
	Personal computer input, output devices.
	Characteristics of the main parts of the laptop
	Electrical Power
	Arithmetic basics of a personal computer
	Logical basics of a personal computer
	Configurations for Specialized Computers
Lecture 3	<b>3. Computer Assembly and Disassembly</b>
	General and Fire Safety
	Install the Motherboard Components
	Install the RAM
	Ports, Connectors, and Cables
	Identify the tools and software used with personal computer components and their purpose.
	Steps of installation work
	Boot the computer after it is assembled
	Computer configuration of the system components
Lecture 4	<b>4. Preventive Maintenance and Troubleshooting</b>
	Personal computer preventive maintenance overview
	Apply Troubleshooting Process to Computer Components and Peripherals
	Setting a computer system in BIOS Setup
	Interaction of the automatic control system with the automatic diagnostic system. POST.
Lecture 5	<b>5. Basic concepts of computer programming</b>
	Features of programming technology

	Object-oriented programming
	Types and composition of programming systems
	A generation of programming languages.
	Programming systems.
	Data Representation in computers. Representing Numbers.
Lecture 6	<b>6. Algorithmization of computational processes</b>
	The concept of algorithm and its main properties.
	Basic concepts of algorithmization of computational processes.
	Variants to set algorithms.
	Structures of algorithms.
	Schemes of algorithms.
	Examples of ways to solve algorithm structures.
	Examples of solving problems for compiling algorithms.
Lecture 7	<b>7. The VBA development environment and its syntax</b>
	The object structure of the VBA language. Main characteristics of components, syntax.
	Basic principles of creating programs in the VBA language.
	Object-oriented programming in VBA.
	Custom classes. Creating object variables.
	Development of the program interface. Components: Form, properties, and methods of the UserForm object; Button, Label, Text Box, and Image.
<b>2 semester</b>	
<b>LECTURES</b>	
Lecture 1	<b>1. Introduction to the C ++ programming language</b>
	General characteristics of language
	Software development technology
	Alphabet and identifiers
	Operations, expressions and operators
	Classification of data types
	Values in C++
	The task of constants
	Existence time and scope of variables
Lecture 2	<b>2. Branch programming</b>
	Development of structured programs
	Conditional instructions: <i>if, else, switch</i>
	Examples of using the if and switch case operators
	Conditional instructions: <i>if then else end if; select case</i>
Lecture 3	<b>3. Loops programming</b>
	The ' <i>while</i> ' loop
	The ' <i>do ... while</i> ' loop
	The <i>for</i> statement

	Operators: <i>for... next; do...loop; while...wend</i>
	Examples of using loop operators.
	Nested loops
	Recommendations for choosing loops
	Control operators in loops
	Examples of using loops
Lecture 4	<b>4. Arrays</b>
	Declaring and initializing arrays
	One-dimensional and two-dimensional arrays
	Examples of using arrays

## LABORATORY WORKS

<i>1 semester</i>	
Laboratory work 1	Basic components of a personal computer in HWINFO64.
Laboratory work 2	Components on the motherboard.
Laboratory work 3	Build a Specialized Computer System.
Laboratory work 4	Diagnostic Software.
Laboratory work 5	Post test. BIOS Setup Utility and Common Trouble Shooting.
Laboratory work 6	A block diagram of algorithms in the MS Office 365 environment.
Laboratory work 7	Execute simple data structure programs and their implementation in the form of projects in the VBA environment. Work with data input and output.
Laboratory work 8	Familiarity with components: Form, Button, Label, Text Box, and Image. Compiling programs using these components.
<i>2 semester</i>	
Laboratory work 1	Introduction to Microsoft Visual C++. Types of projects. Creating a project in Microsoft Visual Studio.
Laboratory work 2	Working with Data Types. Arithmetic and logic operations. Development of algorithms and construction of block diagrams.
Laboratory work 3	Linear programming is based on the C++ language.
Laboratory work 4	Branched programming is based on the C++ language.
Laboratory work 5	Cyclic programming in the C++ language.
Laboratory work 6	Development of the user interface in the Visual Basic language based on branched, cyclical processes.

## PRACTICAL WORKS

Practical work 1	Electricity and Ohms Laws.
Practical work 2	Complete the Computer Assembly.

Practical work 3	Disassemble a Computer.
Practical work 4	Assembly and disassemble the Laptop
Practical work 5	Numbering systems
Practical work 6	Arithmetic operations in positional number system

### **5 Hardware and / or software**

1. A personal computer or laptop with constant access to the Internet.
2. Activated university mail account (student.i.p. @ Nmu.one) at Office365.
3. Active account in the distance education system Moodle.
4. Software:
  - Windows 10;
  - HWINFO64;
  - Internet browser;
  - MS Visual Studio Community 2019;
  - MS Visual Basic;
  - LibreOffice 6.4;
  - MS Office 365;
  - Virtual Desktop;
  - Virtual Laptop;
  - distance learning platform Moodle, MS Teams.

### **6 Knowledge progress testing**

Certification of student achievement is accomplished through transparent procedures based on objective criteria in accordance with the University Regulations “On Evaluation of Higher Education Applicants' Learning Outcomes”.

The level of competencies achieved in relation to the expectations, identified during the control activities, reflects the real result of the student's study of the discipline.

#### **6.1 Grading scales**

Assessment of academic achievement of students of the Dnipro University of Technology is carried out based on a rating (100-point) and institutional grading scales. The latter is necessary (in the official absence of a national scale) to convert (transfer) grades for mobile students.

#### ***The scales of assessment of learning outcomes of the NTUDP students***

Rating	Institutional
90 ... 100	Excellent
74 ... 89	Good
60 ... 73	Satisfactory
0 ... 59	Failed

Discipline credits are scored if the student has a final grade of at least 60 points. A lower grade is considered to be an academic debt that is subject to liquidation in accordance with the Regulations on the Organization of the Educational Process of NTUDP.

## 6.2 Diagnostic tools and evaluation procedures

The content of diagnostic tools is aimed at controlling the level of knowledge, skills, communication, autonomy, and responsibility of the student according to the requirements of the National Qualifications Framework (NQF) up to the 7th qualification level during the demonstration of the learning outcomes regulated by the work program.

During the control activities, the student should perform tasks focused solely on the demonstration of disciplinary learning outcomes (Section 2).

Diagnostic tools provided to students at the control activities in the form of tasks for the intermediate and final knowledge progress testing are formed by specifying the initial data and a way of demonstrating disciplinary learning outcomes.

Diagnostic tools (control tasks) for the intermediate and final knowledge progress testing are approved by the appropriate department.

Type of diagnostic tools and procedures for evaluating the intermediate and final knowledge progress testing are given below.

### **Diagnostic and assessment procedures**

INTERMEDIATE CONTROL			FINAL ASSESSMENT	
training sessions	diagnostic tools	procedures	diagnostic tools	procedures
lectures	control tasks for each topic	task during lectures	comprehensive reference work (CCW)	determining the average results of intermediate controls;  CCW performance during the examination at the request of the student
practical	control tasks for each topic	tasks during practical classes		
	or individual task	tasks during independent work		

During the intermediate control, the lectures are evaluated by determining the quality of the performance of the control specific tasks. Practical classes are assessed by the quality of the control or individual task.

If the content of a particular type of teaching activity is subordinated to several descriptors, then the integral value of the assessment may be determined by the weighting coefficients set by the lecturer.

Provided that the level of results of the intermediate controls of all types of training at least 60 points, the final control can be carried out without the



student's immediate participation by determining the weighted average value of the obtained grades.

Regardless of the results of the intermediate control, every student during the final knowledge progress testing has the right to perform the CDF, which contains tasks covering key disciplinary learning outcomes.

The number of specific tasks of the CDF should be consistent with the allotted time for completion. The number of CDF options should ensure that the task is individualized.

The value of the mark for the implementation of the CDF is determined by the average evaluation of the components (specific tasks) and is final.

The integral value of the CDF performance assessment can be determined by taking into account the weighting factors established by the department for each NLC descriptor.

### 6.3 Evaluation criteria

The actual student learning outcomes are identified and measured against what is expected during the control activities using criteria that describe the student's actions to demonstrate the achievement of the learning outcomes.

To evaluate the performance of the control tasks during the intermediate control of lectures and practicals the assimilation factor is used as a criterion, which automatically adapts the indicator to the rating scale:

$$O_i = 100 a / m,$$

where a - number of correct answers or significant operations performed according to the solution standard; m - the total number of questions or substantial operations of the standard.

Individual tasks and complex control works are expertly evaluated using criteria that characterize the ratio of competency requirements and evaluation indicators to a rating scale.

## 7. Course policy

### 7.1. Academic Integrity Policy.

Academic integrity of students is an important condition for mastering the results of training in the discipline and obtaining a satisfactory grade on the current and final tests. Academic integrity is based on condemnation of the practices of copying (writing with external sources other than those allowed for use), plagiarism (reproduction of published texts by other authors without indication of authorship), fabrication (fabrication of data or facts used in the educational process). The policy on academic integrity is regulated by the Regulation "Regulations on the system of prevention and detection of plagiarism at the Dnipro University of Technology

([http://www.nmu.org.ua/ua/content/activity/us\\_documents/System\\_of\\_prevention\\_and\\_detection\\_of\\_plagiarism.pdf](http://www.nmu.org.ua/ua/content/activity/us_documents/System_of_prevention_and_detection_of_plagiarism.pdf)).

In case of violation of academic integrity by a student (copying, plagiarism, fabrication), the work is evaluated unsatisfactorily and must be repeated. The teacher reserves the right to change the topic of the task.

### **7.2. Communication policy.**

Students must have activated university mail.

It is the student's responsibility to check the mailbox at Office365 once a week (every Sunday).

During the weeks of independent work it is the student's responsibility to work with the distance course "Computer science" ([www.do.nmu.org.ua](http://www.do.nmu.org.ua)).

All written questions to teachers regarding the course should be sent to the university e-mail.

### **7.3. Reassembly policy.**

Works that are submitted in violation of deadlines without good reason are evaluated at a lower grade. Relocation takes place with the permission of the dean's office if there are good reasons (for example, sick leave).

### **7.4. Attending classes.**

Full-time students are required to attend classes. Good reasons for not attending classes are illness, participation in university events, business trips, which must be confirmed by documents in case of prolonged (two weeks) absence. The student must inform the teacher either in person or through the headmaster about the absence from class and the reasons for absence. If a student is ill, we recommend staying home and studying with a distance platform. Students whose health is unsatisfactory and may affect the health of other students will be encouraged to leave the class (such absence will be considered an absence due to illness). Practical classes are not repeated, these assessments cannot be obtained during the consultation. For objective reasons (for example, international mobility), learning can take place remotely - online, in agreement with the teacher.

### **7.5 Evaluation Appeal Policy.**

If the student does not agree with the assessment of his knowledge, he may appeal the assessment made by the teacher in the prescribed manner.

### **7.6. Bonuses.**

Students who regularly attended lectures (have no more than two passes without good reason) and have a written syllabus of lectures receive an additional 2 points to the results of the assessment to the final grade.

### **7.7. Participation in the survey.**

At the end of the course and before the session, students will be asked to fill out anonymously questionnaires (Microsoft Forms Office 365), which will be sent to your university mailboxes. Completing the questionnaires is an important component of your learning activity, which will allow you to assess

the effectiveness of the teaching methods used and take into account your suggestions for improving the content of the discipline "Higher Mathematics".

## **8 Recommended bibliography**

### **1 semester**

1. Standart vyshchoi osvity Ukrainy: pershyi (bakalavrskyi) riven, haluz znan 14 - Elektrychna inzheneriia, spetsialnist 141 - Elektroenerhetyka, elektrotekhnika ta elektromekhanika. – 22 s.

2. Kashtan V.Yu. Methodological instructions for the implementation of laboratory works in the discipline “Computer Engineering and Programming” for students of specialty 141 “Power engineering, electrical engineering and electromechanics” [Electronic resource], Part1. – 2021. URL: [https://it.nmu.org.ua/ua/scientific\\_method\\_materials/teaching\\_materials.php](https://it.nmu.org.ua/ua/scientific_method_materials/teaching_materials.php).

3. Kashtan V.Yu. Computer Engineering and Programming for students of specialty 141 “Power engineering, electrical engineering and electromechanics”, 2021 Moodle. URL: <https://do.nmu.org.ua/course/view.php?id=3446>

4. Osnovy informatyky ta obchysliuvalnoi tekhniky: pidruchnyk / V. H. Ivanov, V. V. Karasiuk, M. V. Hvozdenko; za zah. red. V. H. Ivanova. — Kh.: Pravo, 2015. — 312 s.

5. Sarah L. Harris, David Harris. Digital Design and Computer Architecture: ARM Edition 1st Edition. – Morgan Kaufmann. – 2015. – 584p.

6. Ivanov V. H. Osnovy informatyky ta obchysliuvalnoi tekhniky: pidruch. / V. H. Ivanov, V. V. Karasiuk, M. V. Hvozdenko; zazah. red. V. H. Ivanova. – Kh.: Pravo, 2012.

7. Sommerville I. Software Engineering, 10th ed. — Addison-Wesley / Pearson Education Limited, 2015. — 816 p.

8. Elektronika ta mikroskhemotekhnika: pidruchnyk / O.M. Vorobiova, I.P. Panfilov, M.P. Savytska, Yu.V. Fleita. – Odesa: ONAZ im. O.S. Popova, 2015. – 298 s.

9. Albert Paul Malvino. Digital computer electronics. – New Delhi : Tata Mcgraw Hill Education Pvt. Ltd. – 2011. – 522 p.

10. James Lance. The Beginner's Guide to Engineering: Computer Engineering. - CreateSpace Independent Publishing Platform. – 2013. – 158p. ISBN-10 : 1492981540.

11. Cisco Academy IT Essentials Interactive Tutorial: <https://netacad.com>.

### **2 semester**

1. Bulhakova O. S. Alhorytmizatsiia i prohramuvannia: teoriia ta praktyka : navchalnyi posibnyk dlia dystantsiinoho navchannia / O. S. Bulhakova, V. V. Zosimov, H. V. Khodiakova. – Mykolaiv: SPD Rumiantseva, 2021. – 138 s.

2. Trofymenko O.H. S++. Alhorytmizatsiia ta prohramuvannia : pidruchnyk / O.H. Trofymenko, Yu.V. Prokop, N.I. Lohinova, O.V. Zadereiko. 2-he vyd. pererob. i dopovn. Odesa : Feniks, 2019. 477 s.

3. Roger Mayne. Introduction To Windows And Graphics Programming With Visual C++ (With Companion Media Pack), 2nd Edition. – World Scientific, 2015. – 480 p. ISBN-10: 9814699403, ISBN-13: 978-9814699402.
4. Harvey M. Deitel, Paul J. Deitel. C++ How no Program / 10th Edition. Pearson Education: 2017. 1074 p.
5. Vstup do prohramuvannia movoiu S++. Orhanizatsiia obchyslen: navch. posib. / Yu. A. Bielov, T. O. Karnaukh, Yu. V. Koval, A. B. Stavrovskyi. – K.: Vydavnychopolihrafichnyi tsentr "Kyivskyi universytet", 2012. – 175 s.