RUSSIAN MINISTRY OF SCIENCE AND EDUCATION

FEDERAL STATE BUDGETARY EDUCATIONAL INSTITUTION
OF HIGHER EDUCATION
«BASHKIR STATE UNIVERSITY»

FACULTY OF MATHEMATICS AND INFORMATION TECHNOLOGIES

Approved: at the department meeting
Protocol # 5 from January 26, 2021
Head of the department

_____Yulmukhametov R.S.

Coordinated with:
EMC chairman of the faculty/institute

_____ Efimov A.M.

WORKING PROGRAM OF DISCIPLINE (MODULE)

Discipline Workshop on Computer
(name of the discipline)

Obligatory part
(name of the part enclosing the discipline (obligatory, formed by participants of the educational activity, facultative))

bachelor (undergraduate) program

Course of training (speciality)
01.03.02 Applied mathematics and informatics
(code and name of the course of training (speciality))

Subdivision of the course of training (profile)
Applied programming and data analysis
(name of the profile of training)

Qualification (level of training)
bachelor
(name of the level of training)

Designer (compiler): associate professor of the PEI department, PhD	_____ Garifullin R.N.

For enrollment of: 2021

Ufa 2021

Designer: associate professor, PhD Garifullin Rustem Nailevich.

The working program of the discipline is approved at the meeting of the department of Programming and Economic Informatics,
protocol # 5/1 from January , « 26 » 2021.

Head of the department _____ Yulmukhametov R. S.

The addenda and updates introduced into the working program of the discipline are approved at the meeting of the department of Programming and Economic Informatics,,
protocol # 11 from June , « 15 » 2021.

Head of the department _____ Yulmukhametov R. S.

List of documents and materials

1. List of expected results of education in the discipline correlated with the indicators of reaching the competencies.

As a result of acquiring the educational program the learner should reach the following results in the discipline.

Competency being formed (with code)	Code and name of the indicator of reaching the competency	Learning results in the discipline
GPC-2 – Is able to use and adapt existing mathematical methods and programming systems for the development and implementation of algorithms for solving applied problems.	GPC-2.1 Possesses fundamental knowledge of existing mathematical methods and programming systems for the development and implementation of algorithms for solving applied problems.	Possesses fundamental knowledge of existing programming systems for the development and implementation of algorithms for solving applied problems.
	GPC-2.2. Knows how to use the existing mathematical methods and programming systems for the development and implementation of algorithms for solving applied problems in professional activities.	Knows how to use the existing programming systems for the development and implementation of algorithms for solving applied problems.
	GP-2.3. Has the skills to apply existing mathematical methods and programming systems for the development and implementation of algorithms for solving specific problems.	Has the skills to use programming systems for the development and implementation of algorithms for solving specific problems.
GPC-4 - Is able to solve problems of professional activity using existing information and communication technologies and taking into account the basic requirements of information security	GPC-4.1. Knows the basic existing information and communication technologies to solve problems in the field of professional activity, taking into account the requirements of information security	Knows the basic existing information and communication technologies to solve problems in the field of programming, taking into account the requirements of information security.
	GPC-4.2. Can use existing information and communication technologies to solve problems in the field of professional activity, taking into account the requirements of information security.	Is able to use existing information and communication technologies to solve tasks in the field of programming with taking into account the requirements of information security.
	GPC-4.3. Has the skills to of applying existing information and communication technologies to solve tasks in the field of professional taking into account the requirements information information security requirements.	Has the skills to of applying existing information and communication technologies to solve tasks in the field of programming with taking into account the requirements of information information security.

2. Goal and role of the discipline in the structure of educational program.

The discipline «Workshop on Computer» belongs to the obligatory part.
The discipline is studied in the first and second year, semesters 1-4.
The discipline is closely related to such disciplines as "Languages and methods of programming", "Fundamentals of Computer Science".
The main objectives of the course "Workshop on Computer" - to consolidate and expand the knowledge of computer science and computer science, obtained at school; to teach to use methods and tools to develop programs in high-level languages Python, C#, C++, allowing one to develop professional programming skills. The "Workshop on Computer" course includes study of Microsoft Visual Studio (Rad Studio) programming environment with the purpose of practical implementation of of the developed program product on a PC.
For learning the discipline the background of competencies formed in the previous level of education and verified during the enrollment at the university is required.

3. Content of the working program (duration of the discipline, sorts and forms of classes, educational and methodical support for the individual work of learners).

Content of the working program is given in Appendix 1.

4. Fund of grading materials in the discipline.

4.1. List of competencies and indicators of reaching competencies with expected learning results in the discipline correlated with them. Description of criteria and scales for grading the learning results in the discipline.

GPC-2 – Is able to use and adapt existing mathematical methods and programming systems for the development and implementation of algorithms for solving applied problems..

Code and name of the indicator of reaching the competency	Learning results in the discipline	Grading criteria for learning results	
		«Uncredited»	«Credited»
GPC-2.1 Possesses fundamental knowledge of existing mathematical methods and programming systems for the development and	Possesses fundamental knowledge of existing programming systems for the development and implementation of algorithms for solving applied problems.	Lack of knowledge or fragmented knowledge of existing programming	Generated (possibly incomplete) knowledge of existing programming systems

implementation of algorithms for solving applied problems.		systems to develop and implement algorithms to solve applied problems.	to develop and implement algorithms to solve applied problems.
GPC-2.2. Knows how to use the existing mathematical methods and programming systems for the development and implementation of algorithms for solving applied problems in professional activities.	Knows how to use the existing programming systems for the development and implementation of algorithms for solving applied problems.	Missing skills or fragmented skills to use the apparatus of of existing systems of programming systems to develop and implementation of algorithms for solving applied tasks.	An educated (perhaps unsystematic) ability to Use the apparatus of existing programming systems programming systems for the development and implementation of algorithms of solving applied problems.
GP-2.3. Has the skills to apply existing mathematical methods and programming systems for the development and implementation of algorithms for solving specific problems.	Has the skills to use programming systems for the development and implementation of algorithms for solving specific problems.	Missing or fragmented possession Fragmentary possession skills in applying the apparatus of programming systems for development and implementation algorithms in solving specific tasks.	Successful and systematic (possibly with minor deficiencies) mastery of of application of programming systems skills (with some minor gaps) programming systems for the development and implementation of algorithms for solving specific tasks.

GPC-4 - Is able to solve problems of professional activity using existing information and communication technologies and taking into account the basic requirements of information security

Code and name of the indicator of reaching the competency	**Learning results in the discipline**	**Grading criteria for learning results**	
		«Uncredited»	**«Credited»**
GPC-4.1. Knows the basic existing information and communication technologies to solve problems in the field of professional activity,	Knows the basic existing information and communication technologies to solve problems in the field of programming, taking into account the requirements of information security.	Lack of knowledge or fragmented knowledge of the main existing	Formed (possibly incomplete) knowledge of the basic existing information and

taking into account the requirements of information security		information and communication technologies to solve problems in the field of programming, taking into account information security requirements.	communication technologies to solve problems in the field of programming taking into account the requirements of information security.
GPC-4.2. Can use existing information and communication technologies to solve problems in the field of professional activity, taking into account the requirements of information security.	Is able to use existing information and communication technologies to solve tasks in the field of programming with taking into account the requirements of information security.	Lack of skills or fragmented skills to use existing information and communication technologies to solve problems in the field of programming, taking into account information security requirements.	Formed (possibly unsystematic) ability to use existing information and communication technologies to solve problems in the field of programming, taking into account the requirements of information security.
GPC-4.3. Has the skills to of applying existing information and communication technologies to solve tasks in the field of professional taking into account the requirements information information security requirements.	Has the skills to of applying existing information and communication technologies to solve tasks in the field of programming with taking into account the requirements of information information security.	Absence or fragmentary possession of skills in the use of existing information and communication technologies to solve problems in the field of programming taking into account information security information security.	Successful and systematic (possibly with minor gaps) proficiency in the use of existing information and communication technologies to solve problems in the field of programming, taking into account the requirements of information security.

4.2. Typical grading tasks or other materials required for grading the learning results in discipline correlated with the indicators of reaching the competencies which are set in the educational program. Methodical materials determining the grading procedures for learning results in the discipline.

Code and name of the indicator of reaching the competency	Learning results	Grading tasks
GPC-2.1 Possesses fundamental knowledge of existing mathematical methods and programming systems for the development and implementation of algorithms for solving applied problems.	Possesses fundamental knowledge of existing programming systems for the development and implementation of algorithms for solving applied problems.	Laboratory work, protection of the report on laboratory works.
GPC-2.2. Knows how to use the existing mathematical methods and programming systems for the development and implementation of algorithms for solving applied problems in professional activities.	Knows how to use the existing programming systems for the development and implementation of algorithms for solving applied problems.	Laboratory work, protection of the report on laboratory works.
GP-2.3. Has the skills to apply existing mathematical methods and programming systems for the development and implementation of algorithms for solving specific problems.	Has the skills to use programming systems for the development and implementation of algorithms for solving specific problems.	Laboratory work, protection of the report on laboratory works.

Code and name of the indicator of reaching the competency	Learning results	Grading tasks
GPC-4.1. Knows the basic existing information and communication technologies to solve problems in the field of professional activity, taking into account the requirements of information security	Knows the basic existing information and communication technologies to solve problems in the field of programming, taking into account the requirements of information security.	Laboratory work, protection of the report on laboratory works.

GPC-4.2. Can use existing information and communication technologies to solve problems in the field of professional activity, taking into account the requirements of information security.	Is able to use existing information and communication technologies to solve tasks in the field of programming with taking into account the requirements of information security.	Laboratory work, protection of the report on laboratory works.
GPC-4.3. Has the skills to of applying existing information and communication technologies to solve tasks in the field of professional taking into account the requirements information information security requirements.	Has the skills to of applying existing information and communication technologies to solve tasks in the field of programming with taking into account the requirements of information information security.	Laboratory work, protection of the report on laboratory works.

Criteria for grading are scores which are set by the instructor for various activities (grading tasks) in and upon learning modules listed in the Rating plan of the discipline:

credit - 60 to 110 rating points (including 10 incentive points),

uncredited - from 0 to 59 rating points.

Laboratory work.
Laboratory work 1-10. Semestr 1

1. $z_1 = 2\sin^2(3\pi - 2\alpha)\cos^2(5\pi + 2\alpha)$
   $z_2 = \frac{1}{4} - \frac{1}{4}\sin(\frac{5}{2}\pi - 8\alpha)$.
2. Make a program that prints true, if the point with coordinates (x, y) belongs to the given domain, and false otherwise.
3. Calculate and display as a table the values of the function F on the given interval from a, to b, with step dx.
4. Calculate by help of the series $\ln\frac{x+1}{x-1} = 2\sum_{n=0}^{\infty}\frac{1}{(2n+1)x^{2n+1}} = 2\left(\frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \cdots\right), |x| > 1$ and compare with exact value
5. Compute the part of the series for given: $S = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \cdots + (-1)^n\frac{1}{n}$
6. Form a square matrix of even order n according to the given pattern

$$\begin{pmatrix} 1 & 2 & 3 & 4 & \ldots & n \\ n & n-1 & n-2 & n-3 & \ldots & 1 \\ 1 & 2 & 3 & 4 & \ldots & n \\ n & n-1 & n-2 & n-3 & \ldots & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & 4 & \ldots & n \\ n & n-1 & n-2 & n-3 & \ldots & 1 \end{pmatrix}$$

7. Calculate the sum and the number of positive elements of the matrix A[N, N] above the main diagonal.
8. The words in the message are encrypted - each word is written backwards. Decrypt the message.
9. Check if the number of opening and closing parentheses is the same in the given line.
10. Write and test a function POISK_CH(A, B) that returns a position to the first occurrence of one of a character from string A to string B. If no character of string A is included in string B, return -1.

Laboratory work 1-10. Semestr 2

1. Make a program that contains dynamic information about the availability of buses in the bus fleet.
   The information about each bus includes:
   a. bus number;
   b. the name and initials of the driver;
   c. route number.
   The program shall provide:
   a. Initially generate data on all buses in the fleet as a list;
   b. when each bus leaves the park, the number of the bus is entered, and the program deletes data about this bus from the list of buses

in the park, and writes these data into the list of buses on the route

c. When each bus enters the park, the number of the bus is entered, and the program deletes the data of the bus from the list of buses in the park and posts them to the list of buses in the park;
d. On request, the program gives information about the buses in the park or the buses on the on the route.

2. Describe a structure named STUDENT that contains the following fields:
    a. last name and initials;
    b. group number;
    c. grade level (an array of 5 elements).

   Write a program that does the following:
    a. enter data from the keyboard into an array consisting of ten structures of type STUDENT; the entries order the records should be arranged in ascending order of the group number;
    b. display the names and group numbers of all students in the array if the average of a student is greater than 4.0;
    c. if there are no such students, display a corresponding message.
3. There is a pointer P at the root of a non-empty tree. For each level of the tree, starting from zero, print the sum of the values of the vertices at that level.
4. Recursively describe a logical function equal(T1, T2) that checks for equality of trees T1 and T2.
5. Create a linear unidirectional list of of real numbers. Remove an element from the list before each element with a value between 10 and 20.
6. Make a class description to define one-dimensional arrays of strings of fixed length. Provide for the possibility to access individual array strings by indexes, control for array overruns, perform operations of element-by-element coupling of two arrays to form a new array. to form a new array, to merge two arrays with the exclusion of repeating elements, to display the element of the array by the specified index and the whole array. Write a program that demonstrates working with this class.
7. Fill the sequential access file file_A with integers obtained with the random number generator. Get in file_B those components of file_A that are are even.
8. There is a file containing text written in lowercase Russian letters. Get in another file is given. Get the same text written in uppercase letters.
9. Make a program that inserts a new element F in the list L for each occurrence of element E.
10. Make a program that inserts a new element F in the two-way list L before each occurrence of element E. each occurrence of element E.

Grading criteria (in scores):

For the performance of laboratory work #1-10 (1-2 semestrs)

- 5 points are awarded to the student if there are no comments;
- 3 points deducted to the student if there are insignificant remarks;

- 1 point is awarded to the student if on the whole, the results are correct, but there are

1 point is awarded to the student if the results are generally correct, but there are significant comments.

-0 points a laboratory work is not completed.

For the defense of the report on laboratory works #1-10

-5 points will be assigned to a student if there are no comments;

- 3 points are awarded to the student if there are insignificant remarks;

- 1 point awarded to student if there are significant comments.

- 0 no report.

Laboratory work 1-8. Semestr 3

№1

1. Define the user class according to the task variant .
2. Define the following constructors in the class: without parameters, with parameters, copy.
3. Define a destructor in the class.
4. Define function components in the class for viewing and setting data fields.
5. Define a pointer to a component-function.
6. Define a pointer to an instance of the class.
7. Write a demonstration program in which user class objects are created and destroyed and each constructor and destructor call is accompanied by a corresponding message (which object has called which constructor or destructor).
8. Show the use of an object pointer and a component-function pointer in the program.

Possible cases.

1) Printing. Book printing orders for book production. Production preparation. Designing the electronic image of the book. Volume printing and assembly. Personnel, specialists, locations and salaries.

2) Design of petroleum equipment. Storage of information on projects and designers. Formation of project documents. Personnel, specialists, locations and salaries.

№2.

1. Select the ATD class according to the option.
2. Define and implement constructors, destructors, functions Input (keyboard input) and Print (screen output) in a class, overload assignment operation.
3. Write a class test program and run the test.
4. Supplement the class definition with given overloaded operations (according to the variant).
5. Implement these operations. Carry out testing.

Possible cases.

1. A set with elements of **char** type. Additionally overload the following operations:

   \+ add an item to a set (char + set type);

   \+ the union of sets;

   = = check the sets for equality.

2. A set with elements of **char** type. Additionally overload the following operations:

   - - remove an element from the set (set-char type);
   - the intersection of sets;
   - < a comparison of sets.

№3.

1. Define the class hierarchy (according to the option).
2. Define a static component in the class - a pointer to the beginning of a linked list objects and a static function for viewing the list.
3. Implement classes.
4. Write a demonstration programme in which objects of various classes are created and placed in a list, then the list is viewed.
5. Make the relevant methods non-virtual and see what happens.
6. Implement an option where the object is added to the list when it is created, i.e. in the constructor .

Possible cases.

List of classes:

1. student, teacher, person, head of department;
2. an employee, a person, a worker, an engineer;

№4.

1. Create a template for the target class. Define constructors, destructor the overloaded assignment operation ("=") and the operations specified in the job variant.
2. Write a test program that checks the use of the pattern for standard data types.
3. Perform the test.
4. Define a custom class to be used as a template parameter. Define the necessary functions and overloaded operations in the class.
5. Write a test program that tests the use of a template for a custom type.
6. Perform the test.

List of classes.

1. The class is a one-dimensional array. Additionally overload the following operations:
   * array multiplication; [] index access.

2. The class is a one-dimensional array. Additionally overload the following operations: int() the size of the array;
   [] access by index.

№5.

1. Supplement the class hierarchy of Laboratory Work No. 3 with "group" classes. For example, for the subject area FACULTY, the classes "*faculty*" can be proposed, "*student group*", "*department*". It is recommended to create an abstract class ∀*department*", which will be the ancestor of all groups and an abstract *TObject* class at the head of the whole hierarchy.
2. Write an iterator method for the class-group.
3. Write a procedure or function that is performed on all the objects in the group (see examples in the appendix).
4. Write a demonstration programme that creates, displays and destroys group objects, and demonstrates the use of the iterator.

№6.

1. Define a custom data type (class). Define and implement in it constructors, destructor, assignment, input and output operations for standard threads.
2. Write program #1 to create objects of user class (input initial information from keyboard using overloaded ">>" operation) and save them in a stream (file). The

program must display a message about the number of saved objects and the length of the resulting file in bytes.

3. Carry out a programme test.
4. Implement your own parameter handler for output to the stream.
5. Write a program #2 to read objects from the stream, store them in an array and view the array. To view the objects use overloaded for cout operation << and your own manipulator. Provide a message about the number of read objects and bytes in the program.
6. Execute a programme to read the objects saved in the previous programme from the file and view them.
7. Write programme no. 3 to add objects to the stream.
8. Execute the programme by adding several objects to the stream and view the resulting file.
9. Write programme no. 4 to delete objects from the file.
10. Execute the programme, delete several objects from the stream and view file received.
11. Write programme no. 5 to correct (i.e. replace) the entries in the file.
12. Execute the programme and view the resulting file.

№7.

Write and debug three programmes:

1. The first programme demonstrates the use of container classes for storing embedded data types.
2. The second programme demonstrates the use of container classes to store user-defined data types.
3. The third programme demonstrates the use of STL algorithms.

In programme no. 1, carry out the following

1. Create a container object according to the task variant and fill it with data of the type specified in the task variant.
2. View the container.
3. Modify the container by removing some elements from it and replacing others.
4. View a container using iterators to access its elements.
5. Create a second container of the same class and fill it with the same type of data as the first container.
6. Modify the first container by removing n items from it after the given one and then adding all the items from the second container to it.
7. Review the first and second containers.

In programme no. 2, do the same, but for the user data type.

In programme no. 3, carry out the following

Create a container containing objects of user-defined type. Container type is selected according to the task variant.

1. Sort it in descending order of items.
2. View the container.
3. Using a suitable algorithm, find the element in the container that satisfies the given condition.
4. Move the items satisfying the condition to another (previously empty) container. The type of the second container is determined by the job variant.
5. View the second container.
6. Sort the first and second containers in ascending order of items.
7. View them.
8. Obtain a third container by merging the first two.
9. View the third container.
10. Calculate how many elements satisfying the given condition the third

container contains.

11. Determine if there is an element in the third container that satisfies the given condition.

Possible cases.

№ n/a	First container	Second container	Built-in data type
1	vector	list	Int
2	list	deque	Long

№8.

Supplement laboratory work No. 4. Define and process the exception system for a given set of objects.

Grading criteria (in scores):

For completing laboratory work No. 1-6,8

- 6 marks are awarded to the student if there are no remarks;
- 3 marks are awarded to the student if there are insignificant remarks;
- 1 mark is awarded to a student if the results are generally correct, but there are significant comments.
-0 points laboratory work not completed.

For the defence of the report on laboratory works 1-6,8

-6 points are awarded to the student if there are no remarks;
- 3 marks are awarded to the student if there are insignificant remarks;
- 1 mark is awarded to the student if there are significant observations.
- 0 there is no report.

For completing laboratory work No. 7

- 8 marks are awarded to the student if there are no remarks;
- 5 points are awarded to the student if there are minor remarks;
- 2 marks are awarded to a student if the results are generally correct, but there are significant comments.
- -0 points laboratory work is not completed.

For the defence of the report on laboratory works No. 7

-8 points are awarded to the student if there are no remarks;
-5 points are awarded to the student if there are minor remarks;
- 2 marks are awarded to the student if there are significant observations.
- 0 there is no report.

Laboratory work No. 1-4 Semestr 4

1. Develop and test the Calculator application in a Microsoft Visual Studio environment.
2. Develop and test a "Text Editor" application in Microsoft Visual Studio.
3. Develop and test the Matrix Calculator application in Microsoft Visual Studio.
4. Develop and test a "Graphic Editor" application in the Microsoft Visual Studio environment.

Grading criteria (in scores):

For completing laboratory work No. 1-2

- 12 points are awarded to the student if there are no remarks;

- 7 marks are awarded to the student if there are minor remarks;
- 3 marks are awarded to a student if the results are generally correct, but there are significant comments.
-0 points laboratory work not completed.

For the defence of the report on laboratory work No. 1-2

-12 points are awarded to the student if there are no remarks;
- 8 marks are awarded to a student if there are minor remarks;
-4 marks are awarded to a student if the results are generally correct, but there are significant comments.
- 0 there is no report.

For completing laboratory work No. 3-4

- 13 points are awarded to the student if there are no remarks;
- 7 marks are awarded to the student if there are minor remarks;
- 3 marks are awarded to a student if the results are generally correct, but there are significant comments.
-0 points laboratory work not completed.

For the defence of the report on laboratory work No. 3-4

-13 points are awarded to the student if there are no remarks;
- 8 marks are awarded to a student if there are minor remarks;
-4 marks are awarded to a student if the results are generally correct, but there are significant comments.
- 0 there is no report.

4.3. Rating-plan of the discipline.

Rating-plan of the discipline is given in Appendix 2.

5. Educational, methodic and informational support of the discipline.

5.1. List of references to primary and complementary educational literature necessary for acquiring the discipline.

Primary literature:

1. Think Python: How to Think Like a Computer Scientist ISBN-13: 978-1491939369 .— <URL: https://greenteapress.com/wp/think-python-2e/>.
2. Boris Berezin, Boris Ivanovich. Elementary course of C and C+ : tutorial / B.I. Berezin, S.B. Berezin.- M. : DIALOG-MIFI, 2001 .- 288 p. - Bibliography p.284 .- ISBN 5-86404-075-4. (563 copies).
3. Ivanova G.S. Means of procedural programming Microsoft Visual C ++ 2008 : tutorial / G.S. Ivanova, T.N. Nichushkina, R.S. Samarev ; Bauman Moscow State Technical University. - Moscow : Publishing house of Bauman Moscow State Technical University, 2012. - 140 c. Bibliography: p. 131 ; Proceedings of the University [Electronic resource]. - URL: http://biblioclub.ru/index.php?page=book&id=257648
4. Aleksandrov, E.E. Programming in C in Microsoft Visual Studio 2010 : tutorial / E.E. Aleksandrov, V.V. Afonin. Afonin ; National Open University "INTUIT". - Moscow : Internet-university of Information Technologies, 2010. - 500 c. : ill. ; The same [Electronic resource]. - URL:http://biblioclub.ru/index.php?page=book&id=233564.

Auxiliary literature:

5. Elmanova, N.Z. Introduction to C++ Builder 4.0 / N.Z. Elmanova, S.P. Koshel. - Moscow : Dialogue MEPhI, 2000. - 304 c. Tabl., ill. - ISBN 5-86404-132-7 ; The same [Electronic resource]. - URL: http://biblioclub.ru/index.php?page=book&id=89293
6. Belov, V.V. Programming in Delphi: procedural, object-oriented, visual : textbook for universities / V.V. Belov, V.I. Chistyakova. - 2-nd ed. - Moscow : Hot Line - Telecom, 2014. - 240 c. : tables, schemes, illustration. - Bibliography: p. 231 - ISBN 978-5-9912-0412-5 ; The same [Electronic resource]. - URL: http://biblioclub.ru/index.php?page=book&id=276219.

5.2. List of the Internet resources and software necessary for acquiring the discipline, including professional data bases and reference systems.

1. Electronic Library System "Electronic Library of BashSU" https://elib.bashedu.ru/
2. Electronic Library System "University Library Online" http://www.biblioclub.ru
3. Lan library system https://e.lanbook.com
4. Windows 8 Russian. Windows Professional 8 Russian Upgrade. Contract № 104 of 17.06.2013. Licenses are perpetual.
5. Microsoft Office Standard 2013 Russian. Contract No. 114 of 12.11.2014. Licenses are perpetual.
6. Microsoft Visual Studio Community 2017 development environment (Microsoft Visual Studio Community 2017 software license terms and conditions, free software).
7. AcademicEdition Networked Volume Licenses RAD Studio XE3 Professional Concurrent AppWaveEnglish; agreement #263 dated 07.12.2012.
8. Python IDLE 3.10.1, free software

6. Hardware equipment, materials and rooms necessary for implementing the educational process in the discipline.

Names of specialized rooms, rooms and laboratories	Activity form	Name of the equipment/software
1	***2***	***3***
Computer classrooms 520a, 521, 522, 525, 523, 426, or any other room according to the current time table	*Laboratory work*	1. Windows 8 Russian. Windows Professional 8 Russian Upgrade. 2. Microsoft Visual Studio Community 2017 3. Python IDLE 3.10.1
Library, reading halls, Computer classrooms 426	*Individual work*	Internet.

Appendix 1

FEDERAL STATE BUDGETARY EDUCATIONAL INSTITUTION
OF HIGHER EDUCATION
«BASHKIR STATE UNIVERSITY»

CONTENT OF THE WORKING PROGRAM

of the discipline Workshop on Computer for semesters 1-4

full-time

learning form

Activity	Duration
Total duration of the discipline (CUD / hours)	11/396
Academic hours for the work with instructor	304,8
lectures	
seminars	
laboratory	304
other (consultation in group or individually and other forms of learning activities assuming collaboration of learners with instructor)	0,8
Academic hours for individual work of learners	105,8
Academic hours for preparing to exam/credit test/differentiated credit test (Grading)	91,2

Final grading:
credit in semesters 1,2,3,4

No. n/a	Topic and content	Form of study of materials: lectures, practical classes, seminars, laboratory work, independent work and workload (in hours)				Core and additional literature recommended to students (numbers from the list)	Assignments for students' independent work	Form of ongoing monitoring of progress (colloquia, tests, computer-based tests) tests, etc.)
		LC	PR/SEM	LR	SR			
1	2	3	4	5	6	7	8	9
	1st semester							
1	Basic C++,Pascal language tools. Language Alphabet Identifiers Keywords/Signs Operations.Constants.Comment aria.			10	5,8	1,4	lab report no. 1	Laboratory work, laboratory work reports.
2	Data types.Data type concept.Basic data types Programme structure.			10	6	1,4	lab report no. 1	Laboratory work, laboratory work reports.
3	Variables and expressions. Variables. Operations.Expressions.Output /enter			14	6	1,4	lab report no. 2	Laboratory work, laboratory work reports.
4	Expression operators. Branching operators. Loop operators.Operators transmission of control			14	6	1,4	Laboratory work report no. 2-5	Laboratory work, laboratory work reports.
5	Indexes .References . Arrays .Strings.			12	6	1,4	laboratory report Work No. 6-8	Laboratory work, reports on laboratory work

6	Functions. Declaration and definition of functions. Global variables. Returned value. Function's parameters .Recursive functions. Overloading of functions. Function templates .Function main()			12	6	1,4	lab report no. 9-10	Laboratory work, laboratory work reports.
	Total hours:			72	35,8			0,2
	2nd semester							
1	User-defined data types. Rename types (typedef).Enums (enum) Structures (struct,record). Unions (union)			4		1,4	lab report no. 1-2	Laboratory work, laboratory work reports.
2	Preprocessor directives. The #incluclе directive. The #clefine directive. Conditional compilation directives. The directive #undef.			2		1,4	lab report no. 1-10	Laboratory work, laboratory work reports.
3	Working with files. Writing, reading. Serial access files, text files, binary files. files.			4		1,4	laboratory report №7,8,9,10	Laboratory work, laboratory work reports.
4	Dynamic structures			22		1,4	report on	Laboratory

	Data : Linear Lists : single, double linked.Create, add, delete item, display. Implementing dynamic structures using arrays.						Laboratory work No. 5	work, reports on laboratory work.
5	Dynamic data structures, stacks, queues. Creation, addition, deletion of elements.			20		1,4	laboratory report №7,8,9,10	Laboratory work, laboratory work reports.
6	Dynamic data structures. Binary trees. Search tree, perfectly balanced trees. Creating a tree, adding, removing items. Traversal of a tree.			22		1,4	Laboratory report no. 3,4	Laboratory work, laboratory work reports.
7	Classes.Class Description. Object description. Pointer this. Constructors Copy constructor .Class methods. Static class elements. Static fields. Static methods . Friendly functions and classes . Friendly function . Friendly class . .Destructors			22		1,4	lab report no. 6	Laboratory work, laboratory work reports.

	Total hours:			96	11,8			0,2
	3-semester							
1	Classes. Overloading of operations Overloading of unary operations. Overloading of binary operations . Overloading an assignment operation. Overloading of new and delete operations. Overloading the function call operation.Overloading indexing operations.			8	3,8	1,3,4	lab report no. 1-3	Laboratory work, laboratory work reports.
2	Inheritance.Access keys. Simple inheritance. Virtual methods .Late binding mechanism. Abstract classes.Multiple inheritance.Distinction of structures and associations from classes .			8	4	1,3,4	lab report no. 4	Laboratory work, laboratory work reports.
3	Class Templates.Create Class Templates.Using Class Templates .Template specialisation classes .			8	4	1,3,4	lab report no. 5	Laboratory work, laboratory work reports.
4	Handling of exceptional situations.General mechanism			8	4	1,3,4	laboratory report	Laboratory work, reports on

	exception handling. Exception syntax .						Work No. 8	laboratory work.
5	Exception catching. Function exception list. Exceptions in constructors and destructors . Hierarchies exceptions			8	4	1,3,4	lab report no. 8	Laboratory work, laboratory work reports.
6	Thread classes.Standard threads.Data formatting . Flags and formatting methods Manipulators.Exchange methods with threads.Thread errors . File streams .String Streams.Streams and types defined by the user.			8	4	1,3,4	lab report no. 6	Laboratory work, laboratory work reports.
7	Container classes. Serial containers Vectors (vectors) .Two-way queues (deque) .List (list) .Stack (stack) .Queue (queue) . Priority queues (priority queue) .associative containers.dictionaries (tar).dictionaries with duplicates (multimap). sets (set) sets with duplicates (multiset) .Bit sets (bitset).			8	4	1,3,4	lab report no. 7	Laboratory work, laboratory work reports.

8	Iterators and functional objects.Iterators . Reverse iterators .Insertion iterators. Threaded iterators. Functional objects. Arithmetic functional objects. .Predicates.Negators . Binders .Pointer adapters to functions .Method adapters.			8	4	1,3,4	lab report no. 5	Laboratory work, laboratory work reports.
9	Algorithms. Non-modifying operations with sequences. Modifying operations with sequences. Algorithms associated with sorting.			8	4	1,3,4	lab report no. 7	Laboratory work, laboratory work reports.
	Total hours:			72	35,8			0,2
	4-semester							
1	Creating applications in Microsoft Visual Studio. Creating an MFC application.			8	1,1	1-5	lab report no. 1-4	Laboratory work, laboratory work reports.
2	Working with tests and graphics. Pictures, buttons and cursors in the view window.			8	1,1	1-5	lab report no. 1-4	Laboratory work, laboratory work reports.

3	Menu operation. Adding new items to menus. Changing the operation of menu items. Adding and removing menu items.Adding a context menu.			9	1,1	1-5	lab report no. 1-4	Laboratory work, laboratory work reports.
4	Virtual window, keyboard, child window. Image scaling. Handling of scroll bars Handling of keystrokes. Creating a child window.			9	1,1	1-5	lab report no. 1-4	Laboratory work, laboratory work reports.
5	Basic dialog box controls.Adding a dialog box. Button (Button), CheckBox (CheckBox) Text Box (EditControl). Combo Box Combo Box. List Box Radio Button. Static Text (Static Text) and Group Box (Group Box).Picture (PictureControl). Horizontal scroll bar (HorizontalScrollBar). Slider Control			12	1,2	1-5	lab report no. 1-4	Laboratory work, laboratory work reports.

	Counter (Spin Control). Indicator (Progress Control). Hot Key (Hot Key). List Control. Tree (Tree Control).							
6	Toolbar and status bar. ToolBar. Status bar (StatusBar). Adding buttons to the toolbar. Display and hide a button on the toolbar. Delete and add buttons to the toolbar. Add and remove your toolbar. Add new fields to the status bar. Changing the position and colour of the status bar			9	1,1	1-5	lab report no. 1-4	Laboratory work, laboratory work reports.
7	Working with graphical data. Drawing graphical images. Drawing graphics using a metafile.			9	1,1	1-5	lab report no. 1-4	Laboratory work, laboratory work reports.
	Total hours:			64	7,8			0,2
	Total:			304	91,2			0,8

Appendix 2

Rating-plan of the discipline Workshop on Computer
Course of training (speciality) *01.03.02 Applied mathematics and informatics*
Year 1, semester 1 (fall)

Forms of learning activities of students	Scores for each task	Number of tasks in a module	Scores	
			Minimum	Maximum
Module 1				
Current grading				
Laboratory works	5	5	0	25
Midterm grading				
Defense of the report	5	5	0	25
Module 2				
Current grading				
Laboratory works	5	5	0	25
Midterm grading				
Defense of the report	5	5	0	25
Attending/missing classes (scores for missing classes are subtracted)				
Attending workshops			0	0
TOTAL				100

Rating-plan of the isciline Linear algebra and applications
Course of training (speciality) *01.03.02 Applied mathematics and informatics*
Year 1, semester 2 (spring)

Forms of learning activities of students	Scores for each task	Number of tasks in a module	Scores	
			Minimum	Maximum
Module 1				
Current grading				
Laboratory works	5	5	0	25
Midterm grading				
Defense of the report	5	5	0	25
Module 2				
Current grading				
Laboratory works	5	5	0	25
Midterm grading				
Defense of the report	5	5	0	25
Attending/missing classes (scores for missing classes are subtracted)				
Attending workshops			0	0
TOTAL				100

Rating-plan of the discipline Workshop on Computer
Course of training (speciality) *01.03.02 Applied mathematics and informatics*
Year 2, semester 3 (fall)

Forms of learning activities of students	Scores for each task	Number of tasks in a module	Scores	
			Minimum	Maximum
Module 1				
Current grading				
Laboratory works	5	5	0	25
Midterm grading				
Defense of the report	5	5	0	25
Module 2				
Current grading				
Laboratory works	5	5	0	25
Midterm grading				
Defense of the report	5	5	0	25
Attending/missing classes (scores for missing classes are subtracted)				
Attending workshops			0	0
TOTAL				100

Rating-plan of the isciline Linear algebra and applications
Course of training (speciality) *01.03.02 Applied mathematics and informatics*
Year 2, semester 4 (spring)

Forms of learning activities of students	Scores for each task	Number of tasks in a module	Scores	
			Minimum	Maximum
Module 1				
Current grading				
Laboratory works	5	5	0	25
Midterm grading				
Defense of the report	5	5	0	25
Module 2				
Current grading				
Laboratory works	5	5	0	25
Midterm grading				
Defense of the report	5	5	0	25
Attending/missing classes (scores for missing classes are subtracted)				
Attending workshops			0	0
TOTAL				100