

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 43

УТВЕРЖДАЮ

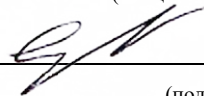
Руководитель направления

д.ф.-м.н., доц.

(должность, уч. степень, звание)

А.О. Смирнов

(инициалы, фамилия)



(подпись)

«15 » июня 2023 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»
(Наименование дисциплины)

Код направления подготовки/ специальности	01.03.02
Наименование направления подготовки/ специальности	Прикладная математика и информатика
Наименование направленности	Прикладная математика и информатика в наукоемком производстве
Форма обучения	очная

Санкт-Петербург– 2023

Лист согласования рабочей программы дисциплины

Программу составил (а)

Ст.преподаватель
(должность, уч. степень, звание)

 12.06.2023
(подпись, дата)

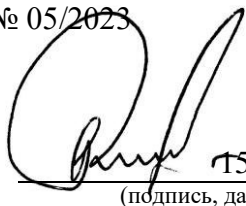
А. А. Фоменкова
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43

«15» июня 2023 г, протокол № 05/2023

Заведующий кафедрой № 43

д.т.н., проф.
(уч. степень, звание)

 15.06.2023
(подпись, дата)

М.Ю. Охтилев
(инициалы, фамилия)

Ответственный за ОП ВО 01.03.02(01)

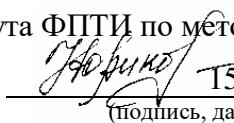
д.ф.-м.н., доц.
(должность, уч. степень, звание)

 15.06.2023
(подпись, дата)

А.О. Смирнов
(инициалы, фамилия)

Заместитель директора института ФПТИ по методической работе

доц., к.ф.-м.н., доц.
(должность, уч. степень, звание)

 15.06.2023
(подпись, дата)

Ю.А. Новикова
(инициалы, фамилия)

Аннотация

Дисциплина «Основы программирования» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 01.03.02 «Прикладная математика и информатика» направленности «Прикладная математика и информатика в наукоемком производстве». Дисциплина реализуется кафедрой «№43».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-2 «Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач»

ОПК-4 «Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности»

ОПК-5 «Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения»

ПК-6 «Способен участвовать в разработке алгоритмического и программного обеспечения средств и систем автоматизации и управления»

Содержание дисциплины охватывает круг вопросов, связанных с изучением структурного и объектно-ориентированных стилей программирования, основных конструкций языка C/C++ и базовыми технологиями создания программ, отвечающих современным требованиям качества и надежности.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, практические занятия, самостоятельную работу обучающегося, курсовое проектирование.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 9 зачетных единиц, 324 часа.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Получение студентами необходимых базовых знаний, принципов и теории, связанных с основами структурного программирования, формирование профессиональной подготовки в области разработки объектно-ориентированных программ, изучение языка программирования C/C++ для разработки прикладных программных средств в задачах профессиональной деятельности, получение навыков разработки, отладки и тестирования программ.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Общепрофессиональные компетенции	ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.3.1 знать математические методы, математические пакеты и системы программирования для разработки и реализации алгоритмов решения прикладных задач
Общепрофессиональные компетенции	ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-4.В.1 владеть навыками разработки алгоритмов решения задач в профессиональной деятельности
Общепрофессиональные компетенции	ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-5.3.1 знать основные алгоритмы и компьютерные программы, пригодные для практического применения при решении задач цифровизации в области профессиональной деятельности ОПК-5.У.1 уметь разрабатывать и применять алгоритмы и компьютерные программы, пригодные для практического применения при решении задач цифровизации в области профессиональной деятельности

		ОПК-5.В.1 владеть практическими навыками разработки и применения алгоритмов и компьютерных программ, пригодных для практического применения при решении задач цифровизации в области профессиональной деятельности
Профессиональные компетенции	ПК-6 Способен участвовать в разработке алгоритмического и программного обеспечения средств и систем автоматизации и управления	ПК-6.3.1 знать языки программирования ПК-6.У.1 уметь разрабатывать алгоритмы и программы, пригодные для практического применения

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

– «Информатика»,

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и могут использоваться при изучении других дисциплин:

– «Объектно-ориентированное программирование»,

– «Алгоритмы и структуры данных»,

– «Операционные системы»

– «Базы данных»,

– «Web-технологии»,

а также в курсовом и дипломном проектировании.

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам	
		№2	№3
1	2	3	4
Общая трудоемкость дисциплины, ЗЕ/ (час)	9/ 324	5/ 180	4/ 144
Из них часов практической подготовки	33	17	16
Аудиторные занятия, всего час.	187	102	85
в том числе:			
лекции (Л), (час)	68	34	34
практические/семинарские занятия (ПЗ), (час)	34	34	
лабораторные работы (ЛР), (час)	68	34	34
курсовой проект (работа) (КП, КР), (час)	17		17
экзамен, (час)	90	54	36
Самостоятельная работа, всего (час)	47	24	23

Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз., Экз.	Экз.	Экз.
---	---------------	------	------

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 2					
Раздел 1. Методологии программирования и этапы разработки программ	4	16	2		2
Раздел 2. Основы работы в C/C++	6	4	13		3
Раздел 3. Функции и их реализация в C/C++	6	6	5		3
Раздел 4. Указатели и ссылки	4	4	2		3
Раздел 5. Массивы	6		4		4
Раздел 6. Строки и их реализация в C/C++	4		4		3
Раздел 7. Основы работы с файлами в C/C++	2	2	4		3
Раздел 8. Пользовательские типы данных в C/C++	2	2			3
Итого в семестре:	34	34	34		24
Семестр 3					
Раздел 9. Основы объектно-ориентированного программирования.	8		8		4
Раздел 10. Перегрузка операций	4		4		4
Раздел 11. Наследование	10		8		4
Раздел 12. Иерархия классов ввода-вывода	4				4
Раздел 13. Обработка исключений	4		6		4
Раздел 14. Шаблоны	4		8		3
Выполнение курсовой работы				17	
Итого в семестре:	34		34	17	23
Итого	68	34	68	17	47

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	Раздел 1. Методологии программирования и этапы разработки программ 1.1. История развития вычислительной техники и языков программирования

	<p>1.2.Этапы разработки программы</p> <p>1.3.Методологии программирования. Структурное программирование</p> <p>1.1. Язык программирования C/C++. Структура программы на языке C.</p>
2	<p>Раздел 2. Основы работы в C/C++</p> <p>2.1. Основные типы данных и их реализация в языке C/C++</p> <p>2.2. Базовые объекты языка C/C++</p> <p>2.3. Операции и выражения в языке C/C++</p> <p>2.4. Консольный ввод и вывод данных в C/C++</p> <p>2.5. Инструкции в C/C++</p>
3	<p>Раздел 3. Функции и их реализация в C/C++</p> <p>3.1. Понятие подпрограмм и особенности их реализации в C/C++.</p> <p>3.2. Объявление и определение функций.</p> <p>3.3. Фактические и формальные параметры функции</p> <p>3.4. Передача параметров в функцию по значению.</p> <p>3.5. Параметры функций со значениями по умолчанию.</p> <p>Перегрузка функций. Функции с переменным числом параметров</p>
4	<p>Раздел 4. Указатели и ссылки</p> <p>4.1. Указатели в языке C/C++.</p> <p>4.2. Ссылки в языке C/C++.</p> <p>4.3. Передача параметров в функцию по указателю и ссылке</p>
5	<p>Раздел 5. Массивы</p> <p>5.1. Определение массива, особенности размещения в памяти</p> <p>5.2. Объявление массива. Инициализация массива</p> <p>Связь массивов и указателей</p> <p>5.4 Одномерные массивы. Типовые алгоритмы обработки массивов</p> <p>5.5 Двумерные массивы. Типовые алгоритмы обработки массивов</p> <p>5.6 Динамическое выделение памяти. Динамические массивы</p> <p>5.7. Передача массивов в функции</p>
6	<p>Раздел 6. Строки и их реализация в C/C++</p> <p>6.1 Представление текстовых данных в ЭВМ. Типовые алгоритмы обработки строк</p> <p>6.2 Функции стандартной библиотеки для работы с символами и строками</p>
7	<p>Раздел 7. Основы работы с файлами в C/C++</p> <p>7.1 Типы файлов и режимы работы.</p> <p>7.2 Консольный и файловый ввод/вывод в языке C/C++</p>
8	<p>Раздел 8. Пользовательские типы данных в C/C++</p> <p>8.1. Перечисления enum</p> <p>8.2. Структуры struct</p>
9	<p>Раздел 9. Основы объектно-ориентированного программирования.</p> <p>9.1.Основные понятия ООП: абстракция, инкапсуляция, наследование, полиморфизм.</p> <p>9.2.Классы. Синтаксис объявления. Модификаторы доступа.</p>

	Сетторы и гетторы. Создание экземпляров классов, ссылки и указатели на объекты. Статические и нестатические члены класса. 9.3.Конструкторы. Назначение конструкторов. Конструктор по умолчанию, копирования, с параметрами. Деструкторы. Назначение деструкторов.
10	Раздел 10. Перегрузка операций 10.1. Механизм перегрузки операций. Синтаксис перегрузки. Правила перегрузки. 10.2. Перегрузка унарной операции. Перегрузка бинарной операции.
11	Раздел 11. Наследование 11.1. Одиночное наследование. Синтаксис объявления производного класса. Управление доступом в производном классе. 11.2. Конструкторы и деструкторы при наследовании. 11.3. Назначение множественного наследования. Ошибка неоднозначности. Проблемы множественного наследования. 11.4. Виртуальные функции и полиморфизм. 11.5. Чисто виртуальные функции. Виртуальные базовые классы. Виртуальное наследование.
12	Раздел 12. Иерархия классов ввода-вывода 12.1. Библиотека ввода и вывода C++. Базовый класс ios. Объекты стандартных устройств. 12.2. Ввод встроенных типов. Вывод встроенных типов. Переопределение операторов ввода и вывода для пользовательских типов.
13	Раздел 13. Обработка исключений 13.1. Механизм обработки исключительных ситуаций. Создание собственных исключений. 13.2. Иерархия классов исключений стандартной библиотеки C++. Создание собственного класса исключений.
14	Раздел 14. Шаблоны 14.1. Шаблоны функций C++. Шаблоны классов. Синтаксис объявления. 14.2. Универсальная инициализация и списки инициализации. Ключевое слово auto. 14.3. Контейнеры и обобщенные алгоритмы. Последовательные контейнеры. Ассоциативные контейнеры.

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 2					
	Блок-схемы алгоритмов.	Решение задач	8	0	1

	Основные обозначения. Разработка алгоритмов решения численных задач в виде блок-схемы				
	Псевдокод. Разработка алгоритмов решения численных задач в виде псевдокода	Решение задач	8	0	1
	Знакомство со средой разработки MS Visual Studio. Создание нового проекта и запуск отладка. Пошаговое выполнение программы в режиме отладки.	Компьютерный практикум	2	0	2
	Библиотека математических функций smath. Особенности выполнения арифметических операций для различных типов данных. Вычисление тригонометрических выражений. Точность вычислений.	Решение задач и компьютерный практикум	2	0	2
	Указатели и ссылки в языке C/C++	Компьютерный практикум	4	0	4
	Объявление и определение функций, область видимости функций. Формальные и фактические параметры функции	Компьютерный практикум	6	0	3
	Хранение числа в виде текстового и бинарного файла	Компьютерный практикум	2		7
	Разработка пользовательского типа данных	Компьютерный практикум	2		8
Всего			34	0	

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№	Наименование лабораторных работ	Трудоемкость,	Из них	№
---	---------------------------------	---------------	--------	---

п/п		(час)	практической подготовки, (час)	раздела дисциплины
Семестр 2				
1	Вводное занятие, инструктаж по технике безопасности	2	0	1
2	Вычисление математических выражений	4	0	2
3	Определение попадания точки в область	4	0	2
4	Программирование поразрядных операций	4	0	2
5	Вычисление кусочной функции	4	0	2,3
6	Обработка числовых последовательностей	4	0	3,4,5
7	Обработка числовых матриц	4	0	3,4,5
8	Обработка текстовых данных	4	0	6
9	Обработка данных в файлах	4	0	7
Семестр 3				
10	Классы. Определение методов класса. Права доступа.	4	4	9
11	Создание класса с конструкторами и деструктором Деструкторы.	4	4	9
12	Перегрузка операций через методы класса. Перегрузка с использованием дружественных функций	4	4	10
13	Одиночное наследование	4	4	11
14	Виртуальные функции и абстрактные классы	4	4	11
15	Обработка исключительных ситуаций	6	6	13
16	Использование обобщенных алгоритмов	8	8	14
Всего		68	34	

4.5. Курсовое проектирование/ выполнение курсовой работы

Цель курсового проекта: выполнение всех этапов разработки, отладки и тестирования программы, а также оформления технической документации по курсовому проекту в соответствии с индивидуальным заданием.

Часов практической подготовки: 10

Примерные темы заданий на курсовую работу приведены в разделе 10 РПД.

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 2, час	Семестр 3, час
1	2	3	4
Изучение теоретического материала дисциплины (ТО)	20	10	10
Подготовка к текущему контролю успеваемости (ТКУ)	7	4	3

Подготовка к промежуточной аттестации (ПА)	20	10	10
Всего:	47	24	23

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий
Перечень печатных и электронных учебных изданий приведен в таблице 8.
Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
https://znanium.com/catalog/product/1007488	Кузин, А. В. Программирование на языке Си : учебное пособие / А.В. Кузин, Е.В. Чумакова.— М. : ФОРУМ : ИНФРА-М, 2019. — 144 с.	-
https://znanium.com/catalog/product/1016471	Дорогов, В. Г. Основы программирования на языке С : учеб. пособие / В.Г. Дорогов, Е.Г. Дорогова ; под ред. проф. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 224 с.	-
https://e.lanbook.com/book/140730	Конова, Е. А. Алгоритмы и программы. Язык С++ : учебное пособие / Е. А. Конова, Г. А. Поллак. — 5-е изд., стер. — Санкт-Петербург : Лань, 2020. — 384 с.	-
https://znanium.com/catalog/product/940363	Культин, Н. Б. С/С++ в задачах и примерах / Н. Б. Культин. — Санкт-Петербург : БХВ-Петербург, 2015. — 285 с. - Текст : электронный.	-
004.4 П 12	С/С++. Программирование на языке высокого уровня : учебник / Т. А. Павловская. - СПб. : ПИТЕР, 2003. - 459 с.	49
004.432-В19	Васильев, А. Н. Самоучитель С++ с примерами и задачами : [учебное пособие] / А. Н. Васильев. - СПб. : Наука и техника, 2010. - 480 с.	Отдел фундаментальной литературы – 30

http://e.lanbook.com/book/1220	Приемы объектно-ориентированного проектирования. Паттерны проектирования = Design Patterns. Elements of reusable object-oriented software/ Э. Гамма [идр.] ;пер. сангл. А. Слинкин. - СПб.: ПИТЕР, 2008. - 366 с. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно ориентированного проектирования. Паттерны проектирования [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2007. — 367с.	
http://e.lanbook.com/book/5115	Ашарина И.В. Объектно-ориентированное программирование в C++: лекции и упражнения. [Электронный ресурс] — Электрон. дан. — М. : "Горячая линия-Телеком", 2012. — 320 с.	

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
https://lms.guar.ru/	Курс "Основы программирования"
https://ravesli.com/uroki-cpp/	Уроки программирования на языке C++
http://cppstudio.com/cat/274/	Язык программирования C++

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
1	Операционная система Microsoft Windows 10
2	Microsoft Visual Studio
3	Adobe Acrobat Reader
4	Офисный пакет Microsoft Office

или Open Office (распространяется свободно)

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Учебная аудитория для проведения лекционных занятий	
2	Учебная аудитория для проведения практических и лабораторных занятий оснащенная специализированной мебелью; техническими средствами обучения, служащими для представления учебной информации большой аудитории; лабораторным оборудованием (ПЭВМ, объединенных в локальную вычислительную сеть с выходом в вычислительную сеть ГУАП и Интернет)	ул. Гастелло, д. 15, лит. А, ауд. 24-03, 24-05; ул. Б.Морская, д. 67, лит. А, ауд. 23-08, 23-09, 23-10

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену; Задачи Тесты
Выполнение курсовой работы	Экспертная оценка на основе требований к содержанию курсовой работы по дисциплине.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	
«отлично» «зачтено»	<ul style="list-style-type: none"> – обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий.
«хорошо» «зачтено»	<ul style="list-style-type: none"> – обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий.
«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> – обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий.
«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> – обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений.

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
1.	Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма	ОПК-2.3.1 ОПК-5.3.1
2.	Методологии программирования и их особенности	ОПК-4.В.1 ОПК-5.3.1
3.	Критерии качества программ	ОПК-4.В.1 ОПК-5.3.1
4.	Этапы разработки программы	ОПК-2.3.1, ОПК-4.В.1
5.	Основные принципы структурного программирования	ОПК-2.3.1 ОПК-5.3.1
6.	Управляющие структуры и их реализация в языке C/C++	ОПК-2.3.1 ПК-6.3.1
7.	Структура программы на языке C/C++	ПК-6.3.1

		ОПК-2.3.1
8.	Простейшие типы данных и их реализация в C/C++	ПК-6.3.1 ОПК-2.3.1
9.	Переменные и константы. Объявление и определение объектов в C/C++	ОПК-2.3.1 ПК-6.3.1
10.	Переменные. Области видимости имен в C/C++	ОПК-2.3.1 ПК-6.3.1
11.	Основные операции и выражения в языке C/C++. Приоритет операций	ОПК-2.3.1 ПК-6.3.1
12.	Основные операции и выражения в языке C/C++. Явное и неявное приведение типов	ОПК-2.3.1 ПК-6.3.1
13.	Функции. Синтаксис объявления и определения функции в языке C/C++. Фактические и формальные параметры функции	ОПК-2.3.1 ПК-6.3.1
14.	Функции. Способы передачи параметров в функцию	ОПК-2.3.1
15.	Указатели в языке C/C++. Адресная арифметика	ОПК-2.3.1 ПК-6.3.1
16.	Массивы. Объявление, инициализация, обращение к элементу	ОПК-2.3.1
17.	Массивы. Связь массивов и указателей	ОПК-2.3.1
18.	Строки в языке C/C++.	ОПК-2.3.1 ПК-6.3.1
19.	Структуры и их реализация в языке C/C++.	ОПК-2.3.1 ПК-6.3.1
20.	Найти действительные корни квадратного уравнения. Проверять действительно ли уравнение квадратное (коэффициент при старшей степени не равен нулю).	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
21.	Выяснить, принадлежит ли точка с координатами (x, y) кругу радиуса r с центром в начале координат	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
22.	Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия и два числа А и В. Выполнить над числами указанное действие и вывести результат.	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
23.	Дано вещественное число А и целое число N (> 0). Вывести все целые степени числа А от 1 до N. Для возведения числа в степень написать функцию самостоятельно (библиотеку <cmath> не использовать)	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
24.	Написать программу, вычисляющую факториал введенного натурального числа.	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
25.	Написать функцию для поиска номера первого минимального элемента массива	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
26.	Дано трёхзначное число x. Напишите оператор присваивания, который в переменную y сохранит сумму цифр числа x. Пример: x=507,y=12.	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
27.	Написать функцию, которая возвращает значение true, если в заданном массиве int M[10] нет нулевых элементов, и false – в противном случае. Привести фрагмент	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1

	программы с объявлением массива и вызовом функции	
28.	Дано число x типа <code>unsigned short</code> . Поменять местами старший и младший байты числа. Ввод-вывод осуществляется в шестнадцатеричной системе счисления. Например: A18F -> 8FA1	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
29.	Вывести на экран заглавные буквы латинского алфавита и их коды в десятичной, шестнадцатеричной и восьмеричной системе счисления	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
30.	Написать программу, которая будет выводить на экран двоичное представление заданного целого числа	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
31.	Напишите функцию, которая меняет местами значения двух целых. В качестве типа параметров используйте <code>int*</code> . Напишите другую функцию с тем же назначением, используя в качестве типа параметров <code>int&</code> .	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
32.	Написать функцию <code>copyN</code> , которая копирует строку в другую строку заданное количество раз	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
33.	Написать функцию, которая выделяет из заданной строки подстроку заданной длины, начиная с заданной позиции	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
34.	ООП – определение. Основные понятия ООП: Абстракция, наследование.	ОПК-2.3.1 ОПК-2.У.1 ПК-6.3.1
35.	ООП – определение. Основные понятия ООП: Инкапсуляция, полиморфизм.	ОПК-2.3.1 ПК-6.3.1
36.	ООП – основные принципы. Определение класса, объекта. Их взаимосвязь между собой.	ОПК-4.У.1 ОПК-4.В.1 ПК-6.3.1
37.	C++ - Пространства имен. Объявление и способы использования.	ОПК-2.У.1 ОПК-4.В.1 ПК-6.У.1
38.	C++ - классы памяти.	ОПК-2.У.1 ОПК-4.В.1 ПК-6.У.1
39.	C++ - Универсальная инициализация. Инициализация по умолчанию базовых типов.	ОПК-2.У.1 ОПК-4.В.1 ПК-6.У.1
40.	Реализация классов в C++. Объявление и определение класса. Данные-члены и функции-члены класса.	ОПК-4.В.1 ПК-6.У.1
41.	Реализация классов в C++. Класс, структура, объединение (<code>class, struct, union</code>) основные отличия.	ОПК-4.В.1 ПК-6.У.1
42.	Реализация классов в C++. Права доступа к членам класса.	ОПК-4.В.1 ПК-6.У.1
43.	Реализация классов в C++. Друзья класса. (Дружественные классы и функции).	ОПК-4.В.1 ПК-6.У.1
44.	Реализация классов в C++. Перегрузка функций-членов.	ОПК-4.В.1 ПК-6.У.1
45.	Реализация классов в C++. Статические и нестатические члены.	ОПК-4.В.1 ПК-6.У.1
46.	Конструкторы. Конструкторы по умолчанию.	ОПК-4.В.1

		ПК-6.У.1
47.	Конструкторы. Инициализирующий конструктор, список инициализации и конструктор копирования.	ОПК-4.В.1 ПК-6.У.1
48.	Конструкторы. Конструктор перемещения.	ОПК-4.В.1 ПК-6.У.1
49.	Деструкторы. Назначение деструктора. Виртуальные деструкторы.	ОПК-4.В.1 ПК-6.У.1
50.	Деструкторы. Использование деструкторов. Порядок уничтожения.	ОПК-4.В.1 ПК-6.У.1
51.	Реализация классов в С++. Указатели на объекты. Указатель this.	ОПК-4.В.1 ПК-6.У.1
52.	Реализация классов в С++. Указатели на члены класса.	ОПК-5.У.1 ОПК-4.В.1 ПК-6.У.1
53.	Реализация классов в С++. Шаблоны классов. Параметризация шаблонов классов.	ОПК-5.У.1
54.	Реализация классов в С++. Шаблоны классов. Специализация шаблонов классов.	ОПК-4.В.1
55.	Реализация классов в С++. Перегрузка операторов.	ПК-6.У.1
56.	Реализация классов в С++. Операторные функции в пространстве имен и как члены класса.	ОПК-5.У.1
57.	Реализация классов в С++. Операторы ввода-вывода.	ОПК-4.В.1
58.	Реализация классов в С++. Перегрузка операторов преобразования типов.	ОПК-6.У.1
59.	Реализация классов в С++. Оператор явного (explicit) преобразования типов.	ОПК-2.3.1 ОПК-5.У.1 ОПК-4.В.1 ПК-6.У.1
60.	Реализация наследования в С++. Одиночное наследование. Область видимости членов класса.	ОПК-5.У.1
61.	Реализация наследования в С++. Множественное и виртуальное наследование.	ОПК-4.В.1
62.	Реализация полиморфизма в С++. Виртуальные функции.	ПК-6.У.1 ОПК-2.3.1
63.	Реализация полиморфизма в С++. Абстрактные классы. Ограничения на использование абстрактных классов.	ОПК-5.У.1
64.	Исключения. Контролируемый блок операций. Операция генерации исключения (throw).	ОПК-4.В.1
65.	Исключения. Реализация обработки исключительных ситуаций. Вложенные контролируемые блоки операций.	ПК-6.3.1
66.	Исключения. Классы стандартной библиотеки для обработки исключений.	ОПК-5.У.1 ОПК-5.В.1
67.	С++ - тип auto.	ОПК-4.В.1
68.	С++ - диапазонный цикл for (...). Синтаксис, применение.	ПК-6.У.1
69.	С++ - Ключевое слово decltype, назначение.	ОПК-5.У.1
70.	С++ - Перечисления с ограниченной областью видимости.	ОПК-4.В.1
71.	Стандартная библиотека С++. Библиотека ввода-вывода. Назначение, основные классы.	ПК-6.У.1
72.	Стандартная библиотека С++. Контейнеры (понятие), принципы использования.	ОПК-5.У.1
73.	Стандартная библиотека С++. Итераторы (понятие),	ОПК-2.3.1

	виды.	ОПК-4.В.1
74.	Стандартная библиотека C++. Последовательные контейнеры.	ОПК-2.3.1 ОПК-5.У.1 ОПК-4.В.1 ПК-6.У.1 ОПК-4.В.1
75.	Стандартная библиотека C++. Ассоциативные контейнеры.	ОПК-2.3.1 ОПК-4.В.1
76.	Стандартная библиотека C++. Неупорядоченные ассоциативные контейнеры.	ОПК-2.3.1 ОПК-4.В.1
77.	Стандартная библиотека C++. Работа со строками символов.	ОПК-2.3.1 ОПК-4.В.1

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.
Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
	Учебным планом не предусмотрено	

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
1	2
1	Предметная область – «Учет успеваемости студентов». Данные о студенте хранятся в структуре с именем STUDENT, содержащей следующие поля: <ul style="list-style-type: none"> • фамилия и инициалы; • номер группы; • успеваемость (массив из пяти элементов). Задание на поиск: найти студентов, чей средний бал не меньше указанного пользователем значения.
2	Предметная область – «Расписание рейсов самолетов». Данные о рейсе хранятся в структуре с именем AEROFLOT, содержащей следующие поля: <ul style="list-style-type: none"> • название пункта назначения рейса; • номер рейса; • тип самолёта. Задание на поиск: найти рейсы, вылетающие в пункт назначения, название которого совпало с названием, введённым с клавиатуры.

3	<p>Предметная область – «Отдел кадров сотрудников». Данные о сотруднике хранятся в структуре WORKER, содержащей следующие поля:</p> <ul style="list-style-type: none"> • фамилию и инициалы работника; • название занимаемой должности; • год поступления на работу. <p>Задание на поиск: найти работников, чей стаж работы в организации превышает значение, введённое с клавиатуры.</p>
4	<p>Предметная область – «Маршруты». Данные о маршруте хранятся в структуре с именем MARSH, содержащей следующие поля:</p> <ul style="list-style-type: none"> • название начального пункта маршрута; • название конечного пункта маршрута; • номер маршрута. <p>Задание на поиск: найти маршрут, номер которого введён с клавиатуры.</p>
5	<p>Предметная область – «Расписание поездов». Данные о маршруте поезда хранятся в структуре TRAIN, содержащей следующие поля:</p> <ul style="list-style-type: none"> • название пункта назначения; • номер поезда; • время отправления. <p>Задание на поиск: найти поезда, отправляющийся после введённого с клавиатуры времени.</p>
6	<p>Предметная область – «Список контактов».</p> <p>Данные о человеке хранятся в структуре с именем NOTE, содержащей следующие поля:</p> <ul style="list-style-type: none"> • фамилия, имя; • номер телефона; • дата рождения (массив из трёх чисел). <p>Задание на поиск: найти информацию о человеке, номер телефона которого введён с клавиатуры.</p>
7	<p>Предметная область – «Информация о сотрудниках».</p> <p>Данные о человеке хранятся в структуре с именем ZNAK, содержащей следующие поля:</p> <ul style="list-style-type: none"> • фамилия, имя; • знак зодиака; • дата рождения (массив из трёх чисел). <p>Задание на поиск: найти информацию о людях, чья фамилия введена с клавиатуры.</p>
8	<p>Предметная область – «Каталог товаров». Данные о товаре хранятся в структуре с именем PRICE, содержащей следующие поля:</p> <ul style="list-style-type: none"> • название товара; • название магазина, в котором продаётся товар; • стоимость товара в рублях. <p>Задание на поиск: найти информации о товаре, название которого введено с клавиатуры.</p>

9	<p>Предметная область – «Платежные поручения». Данные о платеже хранятся в структуре с именем ORDER, содержащей следующие поля:</p> <ul style="list-style-type: none"> • расчётный счёт плательщика; • расчётный счёт получателя; • перечисляемая сумма в рублях. <p>Задание на поиск: найти информацию о сумме, снятой с расчётного счёта плательщика, введённого с клавиатуры.</p>
10	<p>Сгущение таблиц. Линейная интерполяция таблично заданной функции. Функция, заданная в виде координат точек, хранится в массиве структур, каждый элемент которой – структура с полями: аргумент, функция. При добавлении нового элемента программа должна автоматически сортировать массив по возрастанию аргумента. *вывести график решения*</p>
11	<p>Сгущение таблиц. Кубическая интерполяция таблично заданной функции. Функция, заданная в виде координат точек, хранится в виде 2 одномерных массивов: массив значений аргументов и массив значений функции. При добавлении нового элемента программа должна автоматически сортировать массивы по возрастанию аргумента.</p>
12	<p>Вычисление определенного интеграла функции методом трапеции. Функция, заданная в виде координат точек, хранится в списке, каждый элемент которой – структура с полями: аргумент, функция. При добавлении нового элемента программа должна автоматически сортировать список по возрастанию аргумента. *вывести график решения*</p>
13	<p>Аппроксимация таблично заданной функции методом наименьших квадратов. Функция, заданная в виде координат точек, хранится в массиве структур, каждый элемент которой – структура с полями: аргумент, функция. При добавлении нового элемента программа должна автоматически сортировать массив по возрастанию аргумента. *вывести график решения*</p>
14	<p>Интерполяция таблично заданной функции полиномом Лагранжа. Функция, заданная в виде координат точек, хранится в массиве структур, каждый элемент которой – структура с полями: аргумент, функция. При добавлении нового элемента программа должна автоматически сортировать массив по возрастанию аргумента.</p>
15	<p>Интерполяция таблично заданной функции полиномом Ньютона. Функция, заданная в виде координат точек (узлы могут быть распределены неравномерно), хранится в массиве структур, каждый элемент которой – структура с полями: аргумент, функция. При добавлении нового элемента программа должна автоматически сортировать массив по возрастанию аргумента. *вывести график решения*</p>
16	<p>Решение СЛАУ, заданной пользователем. Коэффициенты уравнения хранятся в двумерном динамическом массиве, свободные члены и решение – в одномерных массивах (без использования библиотек matrix и vector). СЛАУ, введенное пользователем должно быть проверено на</p>

17	Численное дифференцирование таблично заданной функции. Функция, заданная в виде координат точек, хранится в массиве структур, каждый элемент которой – структура с полями: аргумент, функция. При добавлении нового элемента программа должна автоматически сортировать массив по возрастанию аргумента. *вывести график решения*
18	Приложение расчета калорийности блюда. В БД №1 «Ингредиенты» (хранится как файл) содержатся калорийности продуктов на 100г . Программа запрашивает ингредиенты для приготовления блюда и их массу, количество порций, вес готового блюда, и рассчитывает калорийность блюда на порцию, на 100г, сохраняет блюдо в отдельную БД №2 «Рецепты» (хранится как файл)
19	Разработать приложение на языке программирования C/C++, выполняющее эффективное кодирование сообщения. На вход программы подается текстовый файл с исходным сообщением, на выходе имеется текстовый файл с закодированным сообщением и соответствующая кодовая таблица.
20	Разработать приложение, которое будет реализовывать расчет синуса угла, введенного пользователем, одним из 5 алгоритмов, выбирая для аргумента наиболее быстродейственный. Приложение должно обеспечить возможность сравнения временной и алгоритмической сложности каждого из алгоритмов для заданного угла.
21	Разработать приложение, которая будет рассчитывать корень трансцендентного уравнения с начальным приближением, заданным пользователем, одним из методов (минимум 3), выбирая для наиболее быстродейственный. Приложение должно обеспечить возможность сравнения временной и алгоритмической сложности каждого из алгоритмов для заданного начального приближения. *уравнение вводится пользователем с клавиатуры

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
1.	<p>Что выведет следующий код при создании экземпляра класса D?</p> <pre> struct A { A(){ cout << "A "; } }; struct B : public A { B() { cout << "B "; } }; struct C : public A { C() { cout << "C "; } }; struct D : B, C { D() { cout << "D "; } </pre>	<p>ОПК-2.3.1 ОПК-2.У.1 ОПК-4.У.1 ОПК-4.В.1</p>

	<pre>}; int main(int argc, char *argv[]) { D d; return 0; }</pre>	
2.	<pre>struct A { ~A(){ cout << "A "; } }; struct B : public A { ~B() { cout << "B "; } }; struct C : public A { ~C() { cout << "C "; } }; struct D : B, C { ~D() { cout << "D "; } }; int main(int argc, char *argv[]) { D d; return 0; }</pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1
3.	<p>Что напечатает следующая программа:</p> <pre>int main() { try { try { cout << 0.5 << " "; throw 2; } catch (int) { cout << "2 "; throw; } catch (char) { } } catch (...) { cout << "? "; } return 0; }</pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1
4.	<p>Каким будет результат выполнения следующего кода:</p> <pre>#include <stdio.h> class Parent { public: void GetValue() { Count(); } private: virtual void Count() { printf("%d", 1); } }; class Child : public Parent { private: void Count() { printf("%d", 2); } }; int main() { Parent * obj = new Child; obj->GetValue(); return 0; }</pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1
5.	<p>Какое значение будет выведено в консоль в результате работы следующей программы:</p> <pre>#include <iostream></pre>	ОПК-2.3.1 ОПК-4.У.1

	<pre>using namespace std; class Base { int* data; public: Base(int size, int value=1) { data = new int(value); } ~Base() { delete data; } Base &operator+=(Base& src) { *data = *data + *src.data; return *this; } operator int() { return *data; } }; int main() { Base a(2); Base b(2, 10); a += b; cout << b << endl; return 0; }</pre>	ОПК-4.В.1
6.	<p>Что необходимо сделать, чтобы программа прошла этап компоновки (редактирования связей) и отработала корректно:</p> <pre>class Counter { public: // 1 void Count(); // 2 }; int main() { Counter obj; obj.Count(); return 0; }</pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1
7.	<p>Что напечатает следующий код:</p> <pre>class B{ public: virtual void msg () {cout << "class B";} }; class D : public B{ public: virtual void msg() {cout << "class D";} }; int main() { B * var = new D; var->msg(); }</pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1
8.	<p>Какое значение будет выведено в консоль в результате работы следующей программы и почему?</p> <pre>#include <iostream> using namespace std; class A { public: void someMethod(double someArg) { cout << "A::someMethod()" << endl; } }; class B : public A { public: void someMethod(int someArg) { cout << "B::someMethod(int) someArg = "<< someArg << endl; } }; int main(int argc, char* argv[]) {</pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1

	<pre> B b; b.someMethod(0.51); return 0; } </pre>	
9.	<p>Что напечатает следующий код:</p> <pre> #include <stdio.h> class Base1 { public: virtual void Count() { printf("%d", 1); } }; class Base2 { public: virtual void Count() { printf("%d", 2); } }; class Child : public Base2, public Base1 { public: void Count() { printf("%d", 3); } }; int main() { Base2 *obj = new Child; obj->Count(); delete obj; return 0; } </pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1
10.	<p>Что напечатает следующий код при создании экземпляра класса X:</p> <pre> #include <iostream> using namespace std; class Y { public: Y() { cout << "Y"; } }; class Z { public: Z() { cout << "Z"; } }; class X : public Z { private: Y m_objY; public: X() { cout << "X"; } }; int main() { X x; return 1; } </pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1
11.	<p>Что необходимо исправить в коде, чтобы он скомпилировался без ошибок:</p> <pre> class A { public: A(void); ~A(int); }; A::A(void) { } void A::~~A(int a) { } int main() { A a; } </pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1

12.	<p>Скомпилируется ли следующий программный код:</p> <pre> class Parent { public: ~Parent() { } virtual void method() { } }; class Child : public Parent { public: Child() { /* захват ресурсов */ } ~Child() { /* освобождение ресурсов */ } void method() { /* программный код */ } }; int main() { Parent * obj = new Child; // программный код delete obj; return 0; } </pre>	<p>ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1</p>
13.	<p>Скомпилируется ли следующая программа (если нет, то почему?):</p> <pre> #include <iostream> using namespace std; class A { public: virtual void printName() = 0; virtual ~A() = 0; private: int i; }; A::~A() {} class B : public A { public: void printName() { cout << "Class B" << endl; } private: int i_b; }; int main() { A objA; B objB; return 0; } </pre>	<p>ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1</p>
14.	<p>Что выведет на печать данный код:</p> <pre> #include <iostream> using namespace std; union X { int a; int b; } x; struct Y { int a; int b; } y; int main(int argc, char *argv[]) { x.a = 10; x.b = 20; y.a = 1; y.b = 2; cout << x.a << " " << x.b << " " << y.a << " " << y.b << endl; } </pre>	<p>ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1</p>

	<pre> return 0; } </pre>	
15.	<p>Скомпилируется ли данный код:</p> <pre> class A { int n; public: int k; }; int main(int argc, char *argv[]) { A a; a.k = 1; a.n = 3.9; return 0; } </pre>	ОПК-2.3.1 ОПК-4.3.1 ОПК-4.У.1 ОПК-4.В.1
16.	<p>Скомпилируется ли данный код:</p> <pre> struct A { int n; public: int k; }; int main(int argc, char *argv[]) { A a; a.k = 1; a.n = 3.9; return 0; } </pre>	ОПК-4.У.1 ОПК-4.В.1
17.	<p>Что выведет на печать данный код:</p> <pre> #include <iostream> using namespace std; void showstat(int curr) { static int n = 1; n += curr; cout << n << " "; } int main() { for (int i = 0; i < 3; i++) showstat(i); return 0; } </pre>	ОПК-4.У.1 ОПК-4.В.1
18.	<p>Что выведет на печать данный код:</p> <pre> #include <iostream> using namespace std; class A { public: static int n; }; int A::n = 0; int main() { A a1; A a2; a1.n = 10; a2.n = 20; } </pre>	ОПК-2.3.1 ОПК-4.У.1 ОПК-4.В.1

	<pre> cout << a1.n << " "; cout << a2.n << endl; return 0; } </pre>	
--	--	--

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
	Не предусмотрено

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

Целью преподавания дисциплины является получение студентами необходимых базовых знаний, принципов и теории, связанных с основами структурного программирования; навыков разработки, отладки и тестирования программ на алгоритмических языках программирования.

11.1. Методические указания для обучающихся по освоению лекционного материала

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

11.2. Методические указания для обучающихся по прохождению практических занятий

Практическое занятие является одной из основных форм организации учебного процесса, заключающаяся в выполнении обучающимися под руководством преподавателя

комплекса учебных заданий с целью усвоения научно-теоретических основ учебной дисциплины, приобретения умений и навыков, опыта творческой деятельности.

Целью практического занятия для обучающегося является привитие обучающимся умений и навыков практической деятельности по изучаемой дисциплине.

Планируемые результаты при освоении обучающимся практических занятий:

- закрепление, углубление, расширение и детализация знаний при решении конкретных задач;
- развитие познавательных способностей, самостоятельности мышления, творческой активности;
- овладение новыми методами и методиками изучения конкретной учебной дисциплины;
- выработка способности логического осмысления полученных знаний для выполнения заданий;
- обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

Темы практических занятий приведены в п. 4.3. Формы практических занятий: компьютерный практикум, разбор типовых задач и решение задач, опросы, контрольные работы.

11.3. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков программирования и тестирования.

Защита лабораторной работы студента осуществляется согласно отчету, в котором должны быть отражены:

- 1) ФИО студента, группа, наименование лабораторной работы, вариант;
- 2) цель работы, задание в соответствии с вариантом
- 3) описание созданных функций;
- 4) текст программы и пример выполнения программы, выводы

Отчет размещается студентом в личном кабинете (в формате *.doc или *.pdf). Результат работы программы представляется лично студентом на занятиях (на компьютере) или в режиме online при дистанционном обучении (с использованием discord, skype, zoom и т.п., по договоренности с преподавателем).

Подробные методические указания по выполнению лабораторных работ и требования к оформлению отчетов приведены в методических указаниях [электронный ресурс кафедры №43 в локальной сети кафедры], путь

\\dcbm\Методическое обеспечение кафедры 43\Основы программирования\

11.4. Методические указания для обучающихся по прохождению курсового проектирования/выполнения курсовой работы

Курсовой проект проводится с целью формирования у обучающихся опыта комплексного решения конкретных задач профессиональной деятельности.

Курсовой проект позволяет обучающемуся: закрепить навыки разработки, тестирования и отладки программы, подготовить пояснительную записку к выполненному проекту.

Подробные методические указания по выполнению курсового проекта, структура и требования к оформлению пояснительной записки приведены в Туманова А.В. Основы программирования. Методические указания к выполнению курсового проекта [электронный ресурс кафедры №43 в локальной сети кафедры], путь

\\dcbm\Методическое обеспечение кафедры 43\Основы программирования \, МУКП_ОП.pdf

11.5. Методические указания для обучающихся по прохождению самостоятельной работы

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся, является учебно-методический материал по дисциплине.

В течение семестра №2 обучающийся должен самостоятельно более глубоко изучить теоретический материал дисциплины с использованием указанной литературы. А также самостоятельно подготовиться к прохождению промежуточной аттестации по дисциплине в форме экзамена.

В течение семестра №3 обучающийся должен в соответствии с заданием на курсовое проектирование самостоятельно разработать программу и подготовить пояснительную записку.

11.6. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

В течение семестра №2 текущий контроль заключается в защите лабораторных работ. Требования к положительному оцениванию текущей успеваемости предусматривают обязательное выполнение всех лабораторных и практических работ в указанные календарные сроки. Результаты текущего контроля успеваемости учитываются при проведении промежуточной аттестации.

В течение семестра №3 текущий контроль заключается в поэтапном выполнении задания на курсовое проектирование в соответствии с графиком.

11.7. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

– экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценивание результатов обучения при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программам высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП».

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой