

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

Кафедра № 43

УТВЕРЖДАЮ
Руководитель образовательной программы
доц., к.т.н.
(должность, уч. степень, звание)

А.А. Фоменкова
(инициалы, фамилия)
(подпись)
«06» февраля 2025 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»
(Наименование дисциплины)

Код направления подготовки/ специальности	09.03.04
Наименование направления подготовки/ специальности	Программная инженерия
Наименование направленности	Проектирование программных систем
Форма обучения	очная
Год приема	2025

Санкт-Петербург– 2025

Лист согласования рабочей программы дисциплины

Программу составил (а)
доцент, канд. техн. наук
(должность, уч. степень, звание) 05.02.2025
(подпись, дата) А.В. Туманова
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43
«06» февраля 2025 г, протокол № 01/2025

Заведующий кафедрой № 43
д.т.н., проф.
(уч. степень, звание) 06.02.2025
(подпись, дата) М.Ю. Охтилев
(инициалы, фамилия)

Заместитель директора института №4 по методической работе
доц., к.т.н.
(должность, уч. степень, звание) 06.02.2025
(подпись, дата) А.А. Фоменкова
(инициалы, фамилия)

Аннотация

Дисциплина «Основы программирования» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.04 «Программная инженерия» направленности «Проектирование программных систем». Дисциплина реализуется кафедрой «№43».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-4 «Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью»

ОПК-6 «Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов»

Содержание дисциплины охватывает круг вопросов, связанных с изучением структурного и объектно-ориентированного 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, самостоятельная работа обучающегося, выполнение курсовой работы.

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 и объектно-ориентированного программирования;
- получение студентам навыков разработки, тестирования и отладки программ на языках высокого уровня;
- формирование общепрофессиональной подготовки в области разработки объектно-ориентированных 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-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.3.1 знает основные стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы ОПК-4.У.1 умеет применять стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы
Общепрофессиональные компетенции	ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.3.1 знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий ОПК-6.У.1 умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ ОПК-6.В.1 имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач

2. Место дисциплины в структуре ОП

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

- информатика,
- основы цифровой грамотности.

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

- алгоритмы и структуры данных,
- объектно-ориентированное программирование,
- проектирование программных систем.

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			
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:	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. Общие сведения	4	8	4		2
Раздел 2. Основные типы данных C/C++, переменные и выражения. Базовые конструкции структурного программирования	10	8	8		4
Раздел 3. Организация и вызов подпрограмм	6	6	6		4
Раздел 4. Указатели и массивы	6	6	8		4
Раздел 5. Строки в C	4	2	4		4
Раздел 6. Файловый ввод-вывод	4	4	4		6
Итого в семестре:	34	34	34		24

Семестр 3					
Раздел 7. Основы объектно-ориентированного программирования.	8		8		2
Раздел 8. Перегрузка операций	4		4		4
Раздел 9. Наследование	10		8		4
Раздел 10. Иерархия классов ввода-вывода	4				2
Раздел 11. Обработка исключений	4		6		4
Раздел 12. Шаблоны	4		8		7
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
Семестр 2	
1	1.1 Общие сведения о языках программирования 1.2 Структура программы на C/C++. Способы описания программы
2	2.1 Типы данных C/C++. Внутреннее представление данных 2.2 Переменные и выражения 2.3 Операция присваивания. Явное и неявное преобразование типов. Выражения. 2.4 Условный оператор и оператор выбора 2.5 Циклы. Операторы передачи управления
3	3.1 Объявление и определение функций. Передача параметров в функцию 3.2 Параметры функций со значениями по умолчанию. Перегрузка функций. Функции с переменным числом параметров 3.3 Функция main. Запуск исполняемого файла из командной строки
4	4.1 Указатели и ссылки. Динамическое выделение памяти C/C++ 4.2 Одномерные и многомерные массивы. 4.3 Динамическое выделение памяти под массивы. Типовые функции обработки массивов
5	5.1 Представление текстовых данных. Типовые алгоритмы обработки строк 5.2 Функции стандартной библиотеки для работы с символами и строками
6	6.1 Типы файлов и режимы работы. Стандартная библиотека ввода-вывода языка C 6.2 Файловые потоки C++

Номер раздела	Название и содержание разделов и тем лекционных занятий
Семестр 3	
7	Основы объектно-ориентированного программирования 7.1 Основные понятия ООП: абстракция, инкапсуляция, наследование, полиморфизм. 7.2 Классы. Синтаксис объявления. Модификаторы доступа. Сетторы и гетторы. Создание экземпляров классов, ссылки и указатели на объекты. Статические и нестатические члены класса. 7.3 Конструкторы. Назначение конструкторов. Конструктор по умолчанию, копирования, с параметрами. Деструкторы. Назначение деструкторов.
8	Перегрузка операций 8.1. Механизм перегрузки операций. Синтаксис перегрузки. Правила перегрузки. 8.2. Перегрузка унарной операции. Перегрузка бинарной операции.
9	Наследование 9.1. Одиночное наследование. Синтаксис объявления производного класса. Управление доступом в производном классе. 9.2. Конструкторы и деструкторы при наследовании. 9.3. Назначение множественного наследования. Ошибка неоднозначности. Проблемы множественного наследования. 9.4. Виртуальные функции и полиморфизм. 9.5. Чисто виртуальные функции. Виртуальные базовые классы. Виртуальное наследование.
10	Иерархия классов ввода-вывода 10.1. Библиотека ввода и вывода C++. Базовый класс ios. Объекты стандартных устройств. 10.2. Ввод встроенных типов. Вывод встроенных типов. Переопределение операторов ввода и вывода для пользовательских типов.
11	Обработка исключений 11.1. Механизм обработки исключительных ситуаций. Создание собственных исключений. 11.2. Иерархия классов исключений стандартной библиотеки C++. Создание собственного класса исключений.
12	Шаблоны 12.1. Шаблоны функций C++. Шаблоны классов. Синтаксис объявления. 12.2. Универсальная инициализация и списки инициализации. Ключевое слово auto. 12.3. 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 дисц.
Семестр 2					
1	Блок-схемы. Основные обозначения. Решение задач с помощью условного оператора	Решение задач	2		1, 2
2	Блок-схемы и алгоритмы. Цикл с параметром	Решение задач	2		1, 2
3	Блок-схемы и алгоритмы. Цикл с предусловием	Решение задач	2		1, 2
4	Блок-схемы и алгоритмы. Цикл с постусловием	Решение задач	2		1, 2
5	Знакомство со средой разработки MS Visual Studio. Создание нового проекта и запуск	Компьютерный практикум	2		1
6	Отладка проекта в среде MS Visual Studio. Пошаговое выполнение программы в режиме отладки. Точки останова	Компьютерный практикум	2		1
7	Библиотека математических функций smath. Особенности выполнения арифметических операций для различных типов данных. Вычисление тригонометрических выражений. Точность вычислений.	Решение задач и компьютерный практикум	2		2
8	Написание библиотеки пользовательских функций. Подключение библиотеки. Прототипы функций. Заголовочные файлы.	Решение задач и компьютерный практикум	2		3
9	Контрольная работа №1	Решение задач, теоретические вопросы	2		1-3
10	Форматированный ввод-вывод. Флаги и форматирующие функции. Манипуляторы. Ввод-вывод данных в различных системах счисления. Особенности выбора типа	Компьютерный практикум	2		1, 2, 6
11	Объявление и определение функций, область видимости функций. Формальные и фактические параметры функции	Компьютерный практикум	2		3

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисц.
12	Статические и динамические массивы. Передача массивов в качестве параметров. Заполнение массива случайными значениями	Компьютерный практикум	2		4
13	Указатели и ссылки. Обращение к элементам массива через указатели. Обращения к переменным через указатели и ссылки.	Решение задач и компьютерный практикум	2		4
14	Типовые функции обработки строк: работа со строками как с массивом символов	Компьютерный практикум	2		5
15	Особенности динамического выделения памяти. Обнаружение утечек памяти.	Компьютерный практикум	2		4, 5
16	Работа с файлами. Различные варианты обработки текстовых файлов	Компьютерный практикум	2		6
17	Контрольная работа №2	Тесты, решение задач	2		1-6
Всего			34		

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисц.
Семестр 2				
1	Вводное занятие, инструктаж по технике безопасности	2		1
2	Вычисление математических выражений	4		1, 2
3	Определение попадания точки в область	4		2, 3
4	Программирование поразрядных операций	4		2
5	Вычисление кусочной функции	4		2, 3
6	Обработка числовых последовательностей	4		3, 4
7	Обработка числовых матриц	4		3, 4
8	Обработка текстовых данных	4		3, 5
9	Обработка данных в файлах	4		3, 6

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисц.
Семестр 3				
1.	Классы. Определение методов класса. Права доступа.	4		7
2.	Создание класса с конструкторами и деструктором Деструкторы.	4		7
3.	Перегрузка операций через методы класса. Перегрузка с использованием дружественных функций	4		8
4.	Одиночное наследование	4		9
5.	Виртуальные функции и абстрактные классы	4		9
6.	Обработка исключительных ситуаций	6		11
7.	Использование обобщенных алгоритмов	8		12
Всего		68		

4.5. Курсовое проектирование/ выполнение курсовой работы

Цель курсовой работы:

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
Изучение теоретического материала дисциплины (ТО)	18	10	8
Курсовое проектирование (КП, КР)	10		10
Расчетно-графические задания (РГЗ)			
Выполнение реферата (Р)			
Подготовка к текущему контролю успеваемости (ТКУ)	7	4	3
Домашнее задание (ДЗ)			
Контрольные работы заочников (КРЗ)			
Подготовка к промежуточной аттестации (ПА)	12	10	2
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/1225391	Дорогов, В. Г. Основы программирования на языке С : учебное пособие / В. Г. Дорогов, Е. Г. Дорогова ; под ред. Л. Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 224 с.	-
https://e.lanbook.com/book/176900	Конова, Е. А. Алгоритмы и программы. Язык C++ : учебное пособие для вузов / Е. А. Конова, Г. А. Поллак. — 6-е изд., стер. — Санкт-Петербург : Лань, 2021. - 384с.	-
https://e.lanbook.com/book/169287	Солдатенко, И. С. Практическое введение в язык программирования Си : учебное пособие / И. С. Солдатенко, И. В. Попов. — Санкт-Петербург : Лань, 2021. — 132 с.	-
004.4 П 12	C/C++. Программирование на языке высокого уровня : учебник / Т. А. Павловская. - СПб. : ПИТЕР, 2003. - 459 с.	49
http://e.lanbook.com/book/1220	Приемы объектно-ориентированного проектирования. Паттерны проектирования = Design Patterns. Elements of reusable object-oriented software/ Э. Гамма [и др.] ; пер. с англ. 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://e.lanbook.com/	Электронная библиотечная система Лань (доступ для обучающихся и работников ГУАП)
https://znanium.com/	Электронная библиотечная система Знаниум (доступ для обучающихся и работников ГУАП)
https://urait.ru/	Образовательная платформа Юрайт (доступ для обучающихся и работников ГУАП)

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	Средства демонстрации PDF файлов
4	Единая электронная образовательная среда ГУАП

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 таблице12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Учебная аудитория для проведения занятий лекционного типа	
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; тесты; задачи
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	
«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
2 семестр		
1	Уровни языков программирования	ОПК-6.3.1
2	Основные этапы разработки программного продукта	ОПК-4.3.1
3	Структурное программирование. Принципы структурного программирования.	ОПК-6.3.1
4	Язык C/C++. Общие сведения и характеристика языка, состав языка	ОПК-6.3.1
5	Алгоритм. Способы записи алгоритма (блок-схемы, псевдокод, ...)	ОПК-4.3.1
6	Структура программы на языке C/C++. Пример простой программы.	ОПК-6.3.1
7	Принципы типизации данных. Иерархия простых типов данных.	ОПК-6.3.1
8	Стандартные типы данных. Таблица характеристик. Особенности выбора типа.	ОПК-6.3.1
9	Явное и неявное преобразование типов. Правила преобразования типов.	ОПК-6.3.1
10	Переменные и константы	ОПК-6.3.1
11	Константы. Специальные символы. Квалификатор const.	ОПК-6.3.1
12	Типы данных, определяемые пользователем. Перечисляемый тип (enum)	ОПК-6.3.1
13	Структуры (назначение, синтаксис, использование, оператор ->)	ОПК-6.3.1
14	Область видимости переменных	ОПК-6.3.1
15	Операции (классификация, особенности записи, таблица приоритетов). Понятие ассоциативности, приоритета, размерности. Операция sizeof.	ОПК-6.3.1
16	Арифметические операции	ОПК-6.3.1
17	Логические операции и операции сравнения	ОПК-6.3.1
18	Поразрядные операции	ОПК-6.3.1
19	Операция присваивания (синтаксис, логика работы, полная и сокращённая форма, порядок выполнения, примеры)	ОПК-6.3.1
20	Тернарная операция ?:. Отличие от условного оператора	ОПК-6.3.1
21	Операторы языка C. Пустой оператор, составной оператор	ОПК-6.3.1
22	Условный оператор. Связь между несколькими условными операторами и сложными логическими выражениями.	ОПК-6.3.1
23	Оператор выбора (синтаксис, логика работы, пример). Оператор break	ОПК-6.3.1
24	Цикл с предусловием. Цикл с постусловием.	ОПК-6.3.1
25	Цикл с параметром. Взаимозаменяемость циклов.	ОПК-6.3.1
26	Операторы передачи управления goto, break и continue	ОПК-6.3.1
27	Указатели	ОПК-6.3.1
28	Ссылки. Различие между указателями и ссылками	ОПК-6.3.1
29	Одномерные массивы. Типовые алгоритмы обработки массивов.	ОПК-6.3.1
30	Двумерные массивы	ОПК-6.3.1
31	Массивы и их связь с указателями. Адресная арифметика	ОПК-6.3.1
32	Передача массивов в качестве параметров функции	ОПК-6.3.1
33	Концепция памяти. Операции для работы с динамической памятью	ОПК-6.3.1
34	Подпрограммы (синтаксис, виды подпрограмм, контекст, пример)	ОПК-6.3.1

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
35	Объявление и определение функций. Оператор return	ОПК-6.3.1
36	Способы передачи параметров в функцию	ОПК-6.3.1
37	Способы передачи значения из одной функции в другую	ОПК-6.3.1
38	Параметры функции со значениями по умолчанию	ОПК-6.3.1
39	Функции с переменным числом параметров	ОПК-6.3.1
40	Перегрузка функций	ОПК-6.3.1
41	Функция main(). Передача параметров в функцию main.	ОПК-6.3.1
42	Представление строк в языках программирования. Строки в C. Основные алгоритмы обработки строк.	ОПК-6.3.1
43	Работа с символами. Основные функции стандартной библиотеки <ctype>.	ОПК-6.3.1
44	Работа со строками. Основные функции стандартной библиотеки <cstring>.	ОПК-6.3.1
45	Файловый ввод-вывод. Стандартная библиотека ввода-вывода (stdio). Типовые алгоритмы обработки файлов.	ОПК-6.3.1
46	Потоковый ввод-вывод. Файловые потоки (fstream). Типовые алгоритмы обработки файлов.	ОПК-6.3.1
47	Потоковый ввод-вывод. Основные функции управления вводом-выводом библиотеки <iostream>.	ОПК-6.3.1
48	Потоковый ввод-вывод. Форматирование данных (функции форматирования и манипуляторы).	ОПК-6.3.1
49	Основные функции управления вводом-выводом <stdio>	ОПК-6.3.1
№ п/п	Перечень задач для экзамена	
1	Определить размер файла в байтах	ОПК-6.У.1
2	Подсчитать число строк в файле	ОПК-6.У.1
3	Скопировать содержимое из первого файла во второй	ОПК-6.У.1
4	Написать функцию, которая убирает из текстового файла все согласные и записывает данные в новый файл	ОПК-6.У.1
5	Дано имя файла. В файле два столбца с числами: найти среднее арифметическое в каждом столбце и вывести результат на экран	ОПК-6.У.1
6	В заданном файле поменять порядок байт на обратный	ОПК-6.У.1
7	В строке найти слово наибольшей длины	ОПК-6.У.1
8	Подсчитать количество символов в файле	ОПК-6.У.1
9	Подсчитать количество слов в файле	ОПК-6.У.1
10	Найти действительные корни квадратного уравнения	ОПК-6.У.1
11	Выяснить, принадлежит ли точка с координатами (x, y) кругу радиуса r с центром в начале координат.	ОПК-4.У.1 ОПК-6.У.1
12	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.У.1 ОПК-6.У.1
13	Дано вещественное число А и целое число N (> 0). Вывести все целые степени числа А от 1 до N. Функцию возведения числа в степень написать без использования библиотеки cmath	ОПК-4.У.1 ОПК-6.У.1
14	Написать программу, вычисляющую факториал введенного числа.	ОПК-4.У.1 ОПК-6.У.1
15	Написать функцию сортировки массива	ОПК-4.У.1 ОПК-6.У.1
16	Дано трёхзначное число x. Напишите оператор присваивания,	ОПК-4.У.1

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
	который в переменную <code>y</code> сохранит сумму цифр числа <code>x</code> . Пример: <code>x=507,y=12</code> .	ОПК-6.У.1
17	Написать функцию, которая возвращает значение <code>true</code> , если в заданном массиве <code>int M[10]</code> нет нулевых элементов, и <code>false</code> – в противном случае.	ОПК-4.У.1 ОПК-6.У.1
18	Дано число <code>x</code> типа <code>unsigned short</code> . Поменять местами старший и младший байты числа. Ввод-вывод осуществляется в шестнадцатеричной системе счисления (например: <code>A18F -> 8FA1</code>)	ОПК-6.У.1
19	Вывести на экран заглавные буквы латинского алфавита и их коды в десятичной, шестнадцатеричной и восьмеричной системе счисления.	ОПК-6.У.1
20	Написать программу, которая будет выводить на экран двоичное представление заданного целого числа	ОПК-4.У.1 ОПК-6.У.1
21	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-6.У.1 ОПК-6.В.1
22	Написать функцию, которая копирует строку в другую строку заданное количество раз	ОПК-4.У.1 ОПК-6.У.1
23	Написать функцию, которая выделяет из заданной строки подстроку заданной длины, начиная с заданной позиции	ОПК-4.У.1 ОПК-6.У.1
3 семестр		
1.	ООП – определение. Основные понятия ООП: Абстракция, наследование.	ОПК-6.3.1
2.	ООП – определение. Основные понятия ООП: Инкапсуляция, полиморфизм.	ОПК-6.3.1
3.	ООП – основные принципы. Определение класса, объекта. Их взаимосвязь между собой.	ОПК-6.3.1
4.	C++ - Пространства имен. Объявление и способы использования.	ОПК-6.3.1
5.	C++ - классы памяти.	ОПК-6.3.1
6.	C++ - Универсальная инициализация. Инициализация по умолчанию базовых типов.	ОПК-6.3.1
7.	Реализация классов в C++. Объявление и определение класса. Данные-члены и функции-члены класса.	ОПК-6.3.1
8.	Реализация классов в C++. Класс, структура, объединение (<code>class</code> , <code>struct</code> , <code>union</code>) основные отличия.	ОПК-6.3.1
9.	Реализация классов в C++. Права доступа к членам класса.	ОПК-6.3.1
10.	Реализация классов в C++. Друзья класса. (Дружественные классы и функции).	ОПК-6.3.1
11.	Реализация классов в C++. Перегрузка функций-членов.	ОПК-6.3.1
12.	Реализация классов в C++. Статические и нестатические члены.	ОПК-6.3.1
13.	Конструкторы. Конструкторы по умолчанию.	ОПК-6.3.1
14.	Конструкторы. Инициализирующий конструктор, список инициализации и конструктор копирования.	ОПК-6.3.1
15.	Конструкторы. Конструктор перемещения.	ОПК-6.3.1
16.	Деструкторы. Назначение деструктора. Виртуальные деструкторы.	ОПК-6.3.1
17.	Деструкторы. Использование деструкторов. Порядок уничтожения.	ОПК-6.3.1
18.	Реализация классов в C++. Указатели на объекты. Указатель <code>this</code> .	ОПК-6.У.1

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
19.	Реализация классов в C++. Указатели на члены класса.	ОПК-6.У.1
20.	Реализация классов в C++. Шаблоны классов. Параметризация шаблонов классов.	ОПК-6.У.1
21.	Реализация классов в C++. Шаблоны классов. Специализация шаблонов классов.	ОПК-6.У.1
22.	Реализация классов в C++. Перегрузка операторов.	ОПК-6.У.1
23.	Реализация классов в C++. Операторные функции в пространстве имен и как члены класса.	ОПК-6.У.1
24.	Реализация классов в C++. Операторы ввода-вывода.	ОПК-6.У.1
25.	Реализация классов в C++. Перегрузка операторов преобразования типов.	ОПК-6.У.1
26.	Реализация классов в C++. Оператор явного (explicit) преобразования типов.	ОПК-6.У.1
27.	Реализация наследования в C++. Одиночное наследование. Область видимости членов класса.	ОПК-6.У.1
28.	Реализация наследования в C++. Множественное и виртуальное наследование.	ОПК-6.У.1
29.	Реализация полиморфизма в C++. Виртуальные функции.	ОПК-6.У.1
30.	Реализация полиморфизма в C++. Абстрактные классы. Ограничения на использование абстрактных классов.	ОПК-6.У.1
31.	Исключения. Контролируемый блок операций. Операция генерации исключения (throw).	ОПК-6.В.1
32.	Исключения. Реализация обработки исключительных ситуаций. Вложенные контролируемые блоки операций.	ОПК-6.У.1
33.	Исключения. Классы стандартной библиотеки для обработки исключений.	ОПК-6.В.1
34.	C++ - тип auto.	ОПК-6.В.1
35.	C++ - диапазонный цикл for (...). Синтаксис, применение.	ОПК-6.В.1 ОПК-6.3.1
36.	C++ - Ключевое слово decltype, назначение.	ОПК-6.В.1
37.	C++ - Перечисления с ограниченной областью видимости.	ОПК-6.В.1
38.	Стандартная библиотека C++. Библиотека ввода-вывода. Назначение, основные классы.	ОПК-6.В.1
39.	Стандартная библиотека C++. Контейнеры (понятие), принципы использования.	ОПК-6.В.1
40.	Стандартная библиотека C++. Итераторы (понятие), виды.	ОПК-6.В.1
41.	Стандартная библиотека C++. Последовательные контейнеры.	ОПК-6.В.1
42.	Стандартная библиотека C++. Ассоциативные контейнеры.	ОПК-6.В.1
43.	Стандартная библиотека C++. Неупорядоченные ассоциативные контейнеры.	ОПК-6.В.1
44.	Стандартная библиотека C++. Работа со строками символов.	ОПК-6.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 выполнения курсовой работы представлен в таблице 17.

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

№ п/п	Примерный перечень тем для выполнения курсовой работы
1	<p>Предметная область – «Учет успеваемости студентов».</p> <p>Данные об одном студенте хранятся в структуре с именем STUDENT, содержащей следующие поля:</p> <ul style="list-style-type: none"> • фамилия и инициалы; • номер группы; • успеваемость (массив из пяти элементов). <p>Список студентов организован в виде линейного списка.</p> <p>Задание на поиск: найти студентов, чей средний бал не меньше указанного пользователем значения.</p>
2	<p>Предметная область – «Расписание рейсов самолетов».</p> <p>Данные о рейсе хранятся в структуре с именем AEROFLOT, содержащей следующие поля:</p> <ul style="list-style-type: none"> • название пункта назначения рейса; • номер рейса; • тип самолета. <p>Перечень рейсов организован в виде линейного списка.</p> <p>Задание на поиск: найти рейсы, вылетающие в пункт назначения, название которого совпало с названием, введенным с клавиатуры</p>
3	<p>Предметная область – «Отдел кадров сотрудников». Данные о сотруднике хранятся в структуре WORKER, содержащей следующие поля:</p> <ul style="list-style-type: none"> • фамилия и инициалы работника; • название занимаемой должности; • год поступления на работу. <p>Перечень сотрудников организован в виде линейного списка.</p> <p>Задание на поиск: найти работников, чей стаж работы в организации превышает значение, введенное 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>Перечень маршрутов организован в виде линейного списка.</p> <p>Задание на поиск: найти маршрут, номер которого введен 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>Перечень поездов организован в виде линейного списка.</p> <p>Задание на поиск: найти поезда, отправляющийся после введенного 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 трех чисел). <p>Перечень контактов организован в виде линейного списка.</p> <p>Задание на поиск: найти информацию о человеке, номер телефона которого введен 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 трех чисел). <p>Перечень сотрудников организован в виде линейного списка.</p> <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 продается товар; • стоимость товара в рублях. <p>Перечень товаров организован в виде линейного списка.</p> <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"> • расчетный счет плательщика; • расчетный счет получателя; • перечисляемая сумма в рублях. <p>Перечень платежных поручений организован в виде линейного списка.</p> <p>Задание на поиск: найти информацию о сумме, снятой с расчетного счета плательщика, введенного с клавиатуры</p>
10	<p>Выполнить сгущение таблично заданной функции, задаваемой пользователем с клавиатуры методом линейной интерполяции. 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> <p>*вывести график решения*</p>
11	<p>Выполнить сгущение таблично заданной функции, задаваемой пользователем с клавиатуры методом кубической интерполяции. 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> <p>*вывести график решения*</p>
12	<p>Вычислить определенный интеграл функции методами прямоугольников, трапеций и сплайнов, и сравнить результаты вычислений. 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.</p> <p>*вывести график решения*</p>

13	<p>Выполнить аппроксимацию таблично заданной функции, задаваемой пользователем с клавиатуры методом наименьших квадратов. Вычислить значение функции в точке, заданной пользователем.</p> <p>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> <p>*вывести график решения</p>
14	<p>Выполнить сгущение таблично заданной функции, задаваемой пользователем с клавиатуры 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> <p>*вывести график решения*</p>
15	<p>Выполнить сгущение таблично заданной функции, задаваемой пользователем с клавиатуры 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>Решить систему линейных алгебраических уравнений (СЛАУ), заданной пользователем в виде матрицы коэффициентов и вектора свободных членов. Коэффициенты уравнения хранятся в двумерном массиве, свободные члены и решение – в одномерных массивах (без использования библиотек matrix и vector). СЛАУ, введенное пользователем должно быть проверено на возможность решения (вырожденность, наличие только 1 решения)</p>
17	<p>Выполнить численное дифференцирование таблично заданной функции, введенной пользователем. 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> <p>*вывести график решения*</p>
18	<p>Приложение расчета калорийности блюда. В БД № 1 «Ингредиенты» (хранится как файл) содержатся калорийности продуктов на 100 г. Программа запрашивает ингредиенты для приготовления блюда и их массу, количество порций, вес готового блюда, и рассчитывает калорийность блюда на порцию, на 100 г, сохраняет блюдо в отдельную БД № 2 «Рецепты» (хранится как файл)</p>
19	<p>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</p>
20	<p>Разработать приложение на языке программирования C/C++, выполняющее сжатие изображения. На вход программы подается файл исходного изображения, на выходе имеется сжатое изображение</p>
21	<p>Разработать приложение, осуществляющее жеребьевку на турнире по каратэ. Приложение должно оперировать со списком участников турнира, позволять добавлять и удалять участников, модифицировать запись об участнике. В качестве результата работы приложение должно выводить «пули» турнирных боев.</p>
22	<p>Разработать приложение, осуществляющее жеребьевку на турнире по шахматам. Приложение должно оперировать со списком участников турнира, позволять добавлять и удалять участников, модифицировать запись об участнике, должно позволять вводить результаты партий и на их основе составлять турнирные таблицы.</p>

23	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.
24	Разработать приложение, которое будет рассчитывать корень трансцендентного уравнения с начальным приближением, заданным пользователем, одним из методов (минимум 3), выбирая наиболее быстродейственный. Приложение должно обеспечить возможность сравнения временной и алгоритмической сложности каждого из алгоритмов для заданного начального приближения. *уравнение вводится пользователем с клавиатуры
25	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). Перечень студентов организован в виде линейного списка. Задание на поиск: найти всех студентов, имеющих оценки 4 и 5
26	Предметная область – «Музыкальный каталог». Данные о треке хранятся в структуре с именем TRACK, содержащей следующие обязательные поля: <ul style="list-style-type: none"> • название; • исполнитель; • жанр; • тип файла. Перечень треков организован в виде линейного списка. Задание на поиск: найти все треки заданного исполнителя *Реализовать возможность доступа к музыкальным файлам через разрабатываемое приложение
27	Предметная область – «Заметки». Разработать приложение на языке программирования C/C++, которое будет организовывать заметки пользователя в виде текстовых файлов. По запросу пользователя приложение должно выводить текст заметки. Приложение должно обеспечивать возможность добавления и удаления заметок, вывода списка заметок по дате создания, а также по теме заметки (несколько заметок могут иметь одну и ту же тему, например «рецепт», «список покупок»...)
28	Предметная область – «Цветочный магазин». Данные о треке хранятся в структуре с именем FLOWER, содержащей следующие обязательные поля: <ul style="list-style-type: none"> • артикул; • название цветка; • цена; • остаток в магазине Каталог цветов организован в виде линейного списка. Приложение должно обеспечить возможность добавления и удаления новых записей, корректировку количества и цены имеющихся в каталоге цветов. Задание на поиск: найти все цветы, цена которых не превышает заданного пользователем значения

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.	Инструкция: выберите один правильный ответ Каким образом реализованы строки в языке программирования С: 1. Массив символов, первый символ которого содержит длину строки 2. Массив символов с маркером окончания строки \0 3. Экземпляр класса <cstring> 4. Экземпляр класса <string>	ОПК-6.3.1												
2.	Инструкция: выберите правильные варианты ответов Что из перечисленного ниже является прототипом функции C/C++? 1. int x(); 2. my_proc(char, char); 3. double fun(int x, y); 4. void f(void); 5. float *f(int x);	ОПК-6.У.1												
3.	Инструкция: установите соответствие между переменными и их значениями. Чему будет равно значение переменных b, c, d после выполнения программы (C/C++)? <pre>#include <iostream> using namespace std; int sum(int a=4, int b=5); int main() { int a = 2, b = 3, c, d; b = sum(a, b); cin >> a; // ввод a=10 c = sum(a); d = sum(); cout<<b<<" " <<c<<" " <<d<<endl; return 0; } int sum(int a, int b){ return a+b;}</pre> <table><tr><td>A</td><td>b</td><td>1</td><td>5</td></tr><tr><td>B</td><td>c</td><td>2</td><td>9</td></tr><tr><td>C</td><td>d</td><td>3</td><td>15</td></tr></table>	A	b	1	5	B	c	2	9	C	d	3	15	ОПК-6.В.1
A	b	1	5											
B	c	2	9											
C	d	3	15											
4.	Инструкция: Запишите соответствующие простейшие типы данных в порядке увеличения памяти, выделяемой под их хранение: 1. double 2. char 3. int	ОПК-6.3.1												
5.	Инструкция: прочитайте текст и запишите развернутый ответ. Что выполняется в следующей строке кода (C/C++)? <pre>typedef int (* A) ();</pre>	ОПК-6.У.1												

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
6.	<p>Инструкция: прочитайте текст и запишите развернутый ответ. Для приведенной ниже функции напишите, что она выполняет? Корректно ли она определена? Приведите пример возможного вызова функции и ее результат.</p> <pre>int func(unsigned int a) { int r = 1; for (unsigned int i = 2; i <= a; i++) r *= i; return r; }</pre>	ОПК-6.В.1
7.	<p>Инструкция: Запишите последовательность основных этапов разработки программного продукта, начиная с начального.</p> <ol style="list-style-type: none"> 1. Кодирование и отладка 2. Анализ предметной области 3. Разработка архитектуры программного обеспечения 4. Тестирование 5. Разработка требований к программному обеспечению 	ОПК-4.3.1
8.	<p>Инструкция: выберите один правильный ответ Этап разработки требований к программному обеспечению завершается, как правило:</p> <ol style="list-style-type: none"> 1. Написанием программы 2. Написанием программы-методики испытаний 3. Составлением технического задания или спецификаций, согласованных с заказчиком 4. Созданием прототипа 	ОПК-4.3.1
9.	<p>Инструкция: выберите правильные варианты ответов. Сценарий использования может оформляться в виде:</p> <ol style="list-style-type: none"> 1. Блок-схемы 2. Таблицы 3. Текстового описания 4. Фрагмента кода 	ОПК-4.3.1
10.	<p>Инструкция: выберите один правильный ответ. Механизм программирования, объединяющий вместе код и данные, которыми он манипулирует называется...</p> <ol style="list-style-type: none"> 1. Наследование 2. Полиморфизм 3. Инкапсуляция 4. Функция 5. Метод 	ОПК-4.3.1
11.	<p>Инструкция: Прочитайте текст и запишите развернутый обоснованный ответ Что такое абстрактный класс в ООП?</p>	ОПК-4.3.1
12.	<p>Инструкция: Прочитайте текст, выберите один правильный ответ Что такое объект в ООП?</p> <ol style="list-style-type: none"> а) Шаблон, описывающий свойства и методы класса б) Экземпляр класса, имеющий конкретные значения свойств в) Метод класса, реализующий определенную функцию г) Переменная, хранящая значение определенного типа данных 	ОПК-4.3.1

№ п/п	Примерный перечень вопросов для тестов	Код индикатора																
13.	Инструкция: Прочитайте текст, выберите правильные варианты ответа Какие из перечисленных модификаторов доступа к членам класса используются в C++? 1 abstract 2 public 3 private 4 protected	ОПК-6.В.1																
13	Инструкция: Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце <table><tr><td>A</td><td>Модификатор доступа, обозначающий защищенные члены класса</td><td>1</td><td>static</td></tr><tr><td>B</td><td>Модификатор доступа, обозначающий статические класса</td><td>2</td><td>public</td></tr><tr><td>C</td><td>Модификатор доступа, обозначающий закрытые члены класса</td><td>3</td><td>protected</td></tr><tr><td>D</td><td>Модификатор доступа, обозначающий открытые члены класса</td><td>4</td><td>private</td></tr></table>	A	Модификатор доступа, обозначающий защищенные члены класса	1	static	B	Модификатор доступа, обозначающий статические класса	2	public	C	Модификатор доступа, обозначающий закрытые члены класса	3	protected	D	Модификатор доступа, обозначающий открытые члены класса	4	private	ОПК-6.У.1
A	Модификатор доступа, обозначающий защищенные члены класса	1	static															
B	Модификатор доступа, обозначающий статические класса	2	public															
C	Модификатор доступа, обозначающий закрытые члены класса	3	protected															
D	Модификатор доступа, обозначающий открытые члены класса	4	private															

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

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;
- получение опыта творческой работы совместно с преподавателем;
- 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.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

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 обучающимися 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.

Требования к проведению практических занятий

Практические задания выполняются студентом лично на занятиях или представляются в режиме online (при дистанционном обучении). В течение семестра студентом должны быть написаны запланированные контрольные работы, которые влияют на результат текущей и промежуточной аттестации.

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 самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

Темы лабораторных работ приведены в разделе 4.4. Перечень заданий приведен методических указаниях по выполнению лабораторных работ. Студент выбирает номер задания в соответствии с вариантом.

Результат работы программы представляется лично студентом на занятиях (на компьютере) или в режиме online (при дистанционном обучении). Защита лабораторной работы студента осуществляется согласно отчету. Отчет может быть принят преподавателем в ЛК только после личной защиты студентом лабораторной работы.

Структура и форма отчета о лабораторной работе

В отчете о лабораторной работе 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

Требования к оформлению отчета о лабораторной работе

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде в файле формата pdf, подготовленном в текстовом редакторе, титульный лист оформлен в соответствии с <https://fs.guap.ru/regdocs/2023/uch/titl/lab.docx>.

Отчет размещается студентом в личном кабинете.

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\,
МЕТОДИЧКА_ОП.pdf.

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

Курсовая работа проводится с целью формирования у обучающихся опыта комплексного решения конкретных задач профессиональной деятельности.

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

Подробные методические указания по выполнению курсовой работы, структура и требования к оформлению пояснительной записки приведены учебном пособии:

А.А. Ключарёв. Основы программирования. Основные этапы разработки программ: учебное пособие / А.А. Ключарев, А.В. Туманова, А.А. Фоменкова ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - Санкт-Петербург : Изд-во ГУАП, 2023.

11.5. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

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 самостоятельно более глубоко изучить теоретический материал дисциплины с использованием указанной литературы и подготовиться к прохождению промежуточной аттестации по дисциплине в форме экзамена; в соответствии с заданием на выполнение курсовой работы 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 успеваемости выполняется по результатам выполнения контрольных работ и защиты лабораторных работ. Требования к положительному оцениванию текущей успеваемости предусматривают обязательное выполнение всех лабораторных, практических и контрольных работ в указанные календарные сроки. Результаты текущего контроля успеваемости учитываются при проведении промежуточной аттестации.

В течение семестра №3 текущий контроль успеваемости выполняется по результатам выполнения и защиты лабораторных работ, а также в выполнении курсовой работы. Результаты текущего контроля успеваемости учитываются при проведении промежуточной аттестации.

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».

К экзамену по дисциплине во 2-м семестре допускается студент, сдавший как минимум 5 лабораторных работ.

Вопросы и задачи для проведения экзамена представлены в таблице 15; примерный перечень вопросов для тестов приведен в таблице 18.

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