

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

Кафедра № 43

УТВЕРЖДАЮ

Ответственный за образовательную
программу

д.ф.-м.н., доц.

(должность, уч. степень, звание)

А.О. Смирнов

(инициалы, фамилия)



(подпись)

«26» июня 2024 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
Год приема	2024

Лист согласования рабочей программы дисциплины

Программу составил (а)

доц., к.т.н.
(должность, уч. степень, звание)

 14.06.24
(подпись, дата)

А.А. Фоменкова
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43

«17» июня 2024 г, протокол № 05/2024

Заведующий кафедрой № 43

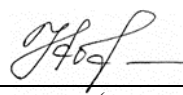
д.т.н., проф.
(уч. степень, звание)

 17.06.2024
(подпись, дата)

М.Ю. Охтилев
(инициалы, фамилия)

Заместитель директора института ФПТИ по методической работе

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

 26.06.24
(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 нахождения рациональных решений в области информационных технологий и систем искусственного интеллекта при разработке организационно-технических процессов
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 на производственных и социальных предприятиях, в некоммерческих организациях, учреждениях социальной сферы и др

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,	Экз.,	Экз.	Экз.

дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз.		
---	------	--	--

Примечание: ** кандидатский экзамен

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	6	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	6	2		3
Раздел 5. Массивы	6		4		4
Раздел 6. Строки и их реализация в C/C++	4	4	4		3
Раздел 7. Основы работы с файлами в C/C++	2	4	4		3
Раздел 8. Пользовательские типы данных в C/C++	2	4			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 1.2. Этапы разработки программы

	<p>1.3.Методологии программирования. Структурное программирование</p> <p>1.4. 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. 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. Массив как структура данных, особенности размещения массивов 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 одномерных массивов</p> <p>5.5 Двумерные массивы. Типовые алгоритмы обработки двумерных массивов</p> <p>5.6 Динамическое выделение памяти под массив</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. Сетторы и гетторы. Создание экземпляров классов, ссылки и</p>

	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. Основные	Решение задач	2	0	1

	обозначения. Разработка алгоритмов решения численных задач в виде блок-схемы				
	Псевдокод. Разработка алгоритмов решения численных задач в виде псевдокода. Метод пошаговой детализации	Решение задач	4	0	1
	Работа в IDE Visual Studio. Отладка. Тестирование	Компьютерный практикум	4	0	2
	Указатели и ссылки в языке C/C++	Компьютерный практикум	6	0	4
	Объявление и определение функций, область видимости функций. Формальные и фактические параметры функции	Компьютерный практикум	6	0	3
	Обработка символьных строк (С-строк) в языке C/C++	Компьютерный практикум и решение задач	4		6
	Хранение числа в виде текстового и бинарного файла	Компьютерный практикум	4		7
	Разработка пользовательского типа данных на языке программирования C/C++	Компьютерный практикум	4		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

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
-------	---------------------------------	---------------------	---------------------------------------	----------------------

Семестр 2				
1	Вводное занятие, инструктаж по технике безопасности	2	0	1
2	Вычисление математических выражений	4	2	2
3	Определение попадания точки в область	4	2	2
4	Программирование поразрядных операций	4	2	2
5	Вычисление кусочной функции	4	2	2,3
6	Обработка числовых последовательностей	4	2	3,4,5
7	Обработка числовых матриц	4	2	3,4,5
8	Обработка текстовых данных	4	2	6
9	Обработка данных в файлах	4	3	7
Семестр 3				
10	Определение и реализация класса с закрытыми данными с использованием сеттеров и геттеров	4	1	9
11	Определение и реализация класса с конструкторами и деструктором	4	1	9
12	Определение класса с использованием перегруженных операторов (перегруженный оператор – член класса, перегруженный оператор – является дружественным)	4	1	10
13	Определение и реализация иерархии классов (наследование)	4	1	11
14	Определение и реализация иерархии классов с использованием виртуальных функций и абстрактных классов	4	1	11
15	Определение класса, содержащего обработку исключительных ситуаций	6	1	13
16	Определение и реализация класса с использованием шаблона и обобщенных алгоритмов	8	1	14
Всего		68	24	

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: 9

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	5
Выполнение курсовой работы			10
Подготовка к текущему контролю успеваемости (ТКУ)	7	4	3
Подготовка к промежуточной аттестации (ПА)	20	10	5
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	Отдел фундаментальн

	пособие] / А. Н. Васильев. - СПб. : Наука и техника, 2010. - 480 с.	ой 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 С++: лекции и упражнения. [Электронный ресурс] — Электрон. дан. — М. : "Горячая линия-Телеком", 2012. — 320 с.	
https://lib.guap.ru/jirbis2/components/irbis/pdf_view/?425385	Ключарев, Александр Анатольевич. 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, 2024. - 96 с. : рис., табл. - Библиогр.: с. 70 (13 назв.). - ISBN 978-5-8088-1919-1 : 105.85 р.	

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.guap.ru/	Курс "Основы программирования"
https://ravesli.com/uroki-cpp/	Уроки программирования на языке С++
http://cppstudio.com/cat/274/	Язык программирования С++

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 таблице12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	
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 5-балльная шкала	Характеристика сформированных компетенций
«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.	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 A и B. Выполнить над числами указанное действие и вывести результат.	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
23.	Дано вещественное число A и целое число N (> 0). Вывести все целые степени числа A 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	ОПК-4.В.1

	присваивания, который в переменную у сохранит сумму цифр числа x. Пример: x=507,y=12.	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 программы с объявлением массива и вызовом функции	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
28.	Дано число x типа unsigned short. 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 int*. Напишите другую функцию с тем же назначением, используя в качестве типа параметров int&.	ОПК-4.В.1 ОПК-5.В.1 ПК-6.У.1
32.	Написать функцию сорuN, 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
3 семестр		
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 (class, struct, union) 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.	ОПК-4.В.1

	(Дружественные классы и функции).	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 C++. Указатели на объекты. Указатель this.	ОПК-4.В.1 ПК-6.У.1
52.	Реализация классов в C++. Указатели на члены класса.	ОПК-5.У.1 ОПК-4.В.1 ПК-6.У.1
53.	Реализация классов в C++. Шаблоны классов. Параметризация шаблонов классов.	ОПК-5.У.1
54.	Реализация классов в C++. Шаблоны классов. Специализация шаблонов классов.	ОПК-4.В.1
55.	Реализация классов в C++. Перегрузка операторов.	ПК-6.У.1
56.	Реализация классов в C++. Операторные функции в пространстве имен и как члены класса.	ОПК-5.У.1
57.	Реализация классов в C++. Операторы ввода-вывода.	ОПК-4.В.1
58.	Реализация классов в C++. Перегрузка операторов преобразования типов.	ОПК-6.У.1
59.	Реализация классов в C++. 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 C++. Одиночное наследование. Область видимости членов класса.	ОПК-5.У.1
61.	Реализация наследования в C++. Множественное и виртуальное наследование.	ОПК-4.В.1
62.	Реализация полиморфизма в C++. Виртуальные функции.	ПК-6.У.1 ОПК-2.3.1
63.	Реализация полиморфизма в C++. 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.	C++ - тип auto.	ОПК-4.В.1
68.	C++ - диапазонный цикл for (...). Синтаксис, применение.	ПК-6.У.1
69.	C++ - Ключевое слово decltype, назначение.	ОПК-5.У.1

70.	C++ - Перечисления с ограниченной областью видимости.	ОПК-4.В.1
71.	Стандартная библиотека C++. Библиотека ввода-вывода. Назначение, основные классы.	ПК-6.У.1
72.	Стандартная библиотека C++. Контейнеры (понятие), принципы использования.	ОПК-5.У.1
73.	Стандартная библиотека C++. Итераторы (понятие), виды.	ОПК-2.3.1 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	Предметная область – «Учет успеваемости студентов». Данные об одном 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). Список студентов организован в виде линейного списка. 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 самолета. Перечень рейсов организован в виде линейного списка. Задание на поиск: найти рейсы, вылетающие в пункт назначения, название которого совпало с названием, введенным 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"> • фамилия и инициалы работника; • название занимаемой должности; • год поступления на работу. <p>Перечень сотрудников организован в виде линейного списка. 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. 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. 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. 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. 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;

	<ul style="list-style-type: none"> • стоимость товара в рублях. <p>Перечень товаров организован в виде линейного списка. 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. 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. *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. *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. *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. 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>Выполнить сгущение таблично заданной функции, задаваемой пользователем с клавиатуры полиномом Лагранжа. Функция, заданная в виде координат точек, хранится в массиве структур,</p>

	<p>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.</p> <p>*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 <code>matrix</code> и <code>vector</code>). 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,</p>

	модифицировать запись об участнике. В качестве результата работы приложение должно выводить «пули» турнирных боев.
22	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.
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	<p>Предметная область – «Цветочный магазин».</p> <p>Данные о треке хранятся в структуре с именем FLOWER, содержащей следующие обязательные поля:</p> <ul style="list-style-type: none"> • артикул; • название цветка; • цена; • остаток в магазине <p>Каталог цветов организован в виде линейного списка.</p> <p>Приложение должно обеспечить возможность добавления и удаления новых записей, корректировку количества и цены имеющихся в каталоге цветов.</p> <p>Задание на поиск: найти все цветы, цена которых не превышает заданного пользователем значения</p>
29	<p>Разработать приложение «Трекер полезных привычек». Приложение должно обеспечивать:</p> <p>Хранение предопределенного набора полезных привычек;</p> <p>Возможность пользователю добавления и удаления полезных привычек из перечня;</p> <p>Возможность ставить цели по достижению полезных привычек: выбор привычки из перечня, установление сроков, отметка о достижении</p>

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

Таблица 18 – Примерный перечень вопросов для тестов

Номер	Содержание теста	Ключ ответа	Код индикатора
1.	<p>Инструкция: Прочитайте текст, выберите правильные варианты ответа</p> <p>Интегрированная среда разработки (IDE) – это программное средство или комплекс программных средств, обязательно предоставляющее следующие возможности</p> <ol style="list-style-type: none"> 1. Имеет редактор программного кода 2. Поддерживает различные языки программирования 3. Средства запуска программного кода 4. Имеет инструменты для отладки программного кода 5. Имеет комплекс встроенных библиотек заданного языка программирования 	1,3,4	ОПК-2.3.1
2.	<p>Инструкция: дайте определение понятию «Жизненный цикл программного обеспечения»</p>	Жизненный цикл программного обеспечения — период времени, который начинается с момента	ОПК-2.3.1

		принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.																	
3.	<p>Инструкция: Для каждого языка программирования, указанного в левом столбце, выберите соответствующий основной тип трансляции, указанный в правом столбце</p> <table border="1"> <tr> <td>A</td> <td>C/C++</td> <td>1</td> <td>компиляция</td> </tr> <tr> <td>B</td> <td>MATLAB</td> <td>2</td> <td>интерпретация</td> </tr> <tr> <td>C</td> <td>Python</td> <td></td> <td></td> </tr> <tr> <td>D</td> <td>Java Script</td> <td></td> <td></td> </tr> </table>	A	C/C++	1	компиляция	B	MATLAB	2	интерпретация	C	Python			D	Java Script			A1,B2,C2,D2	ОПК-2.3.1 , ПК-6.У.1
A	C/C++	1	компиляция																
B	MATLAB	2	интерпретация																
C	Python																		
D	Java Script																		
4.	<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 	3	ОПК-2.3.1																
5.	<p>Инструкция: Прочитайте текст и запишите развернутый обоснованный ответ Что такое абстрактный класс в ООП</p>	Абстрактный класс - это класс, который содержит хотя бы один чисто виртуальный метод и не может быть создан в качестве объекта	ОПК-2.3.1																
6	<p>Инструкция: Прочитайте текст, выберите один правильный ответ</p> <p>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 	б) Экземпляр класса, имеющий конкретные значения свойств																	

	типа данных		
7.	<p>Инструкция: Прочитайте текст, выберите один правильный ответ</p> <p>Выберите корректное определение интерпретации:</p> <ol style="list-style-type: none"> 1. Интерпретация - это - преобразование исходного кода программы с какого-либо языка программирования на машинный язык. 2. Интерпретация - это построчный анализ, обработка и выполнение исходного кода программы или запроса 3. Интерпретация - это комплекс программных средств, используемый программистами для разработки программного обеспечения 	2	ОПК-4.В.1
8.	<p>Инструкция: Прочитайте текст, выберите один правильный ответ</p> <p>Выберите корректное определение компиляции:</p> <ol style="list-style-type: none"> 1. Компиляция – это преобразование исходного кода программы с какого-либо языка программирования на машинный язык. 2. Компиляция – это построчный анализ, обработка и выполнение исходного кода программы или запроса 3. Компиляция – это комплекс программных средств, используемый программистами для разработки программного обеспечения 	1	ОПК-4.В.1
9.	<p>Инструкция: Проанализируйте фрагмент программного кода. В ответе запишите значение переменной k , которое будет получено в результате работы этого кода:</p> <pre>int i, k=0; do { k= k-2; } while (k>0);</pre>	-2	ОПК-4.В.1
10.	<p>Инструкция: Проанализируйте фрагмент программного кода. В ответе запишите значение переменной k , которое будет получено в результате работы этого кода:</p> <pre>int i, k=0; while (k>0) { k = k-2; }</pre>	0	ОПК-4.В.1
11.	<p>Инструкция: Проанализируйте фрагмент программного кода. В ответе запишите значение переменной k , которое будет получено в результате работы этого кода:</p>	6	ОПК-4.В.1

	<pre>int i, k=0; while(k<5) { k= k+2; }</pre>																		
12.	<p>Инструкция: Прочитайте текст, выберите правильные варианты ответа</p> <p>Какие из перечисленных модификаторов доступа к членам класса используются в C++?</p> <p>1 abstract 2 public 3 private 4 protected</p>	234																	
13.	<p>Инструкция: Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце</p> <table border="1" data-bbox="344 779 987 1684"> <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 члены класса</td> <td>2</td> <td>public</td> </tr> <tr> <td>C</td> <td>Модификатор доступа, обозначающий закрытые члены класса</td> <td></td> <td>protected</td> </tr> <tr> <td>D</td> <td>Модификатор доступа, обозначающий открытые члены класса</td> <td></td> <td>private</td> </tr> </table>	A	Модификатор доступа, обозначающий защищенные члены класса	1	static	B	Модификатор доступа, обозначающий статические члены класса	2	public	C	Модификатор доступа, обозначающий закрытые члены класса		protected	D	Модификатор доступа, обозначающий открытые члены класса		private	A3,B1,C4,D2	ОПК-2.3.1
A	Модификатор доступа, обозначающий защищенные члены класса	1	static																
B	Модификатор доступа, обозначающий статические члены класса	2	public																
C	Модификатор доступа, обозначающий закрытые члены класса		protected																
D	Модификатор доступа, обозначающий открытые члены класса		private																
14.	<p>Инструкция: Прочитайте текст, выберите один правильный ответ</p> <p>Выберите корректное определение алгоритма:</p> <ol style="list-style-type: none"> Алгоритм — это предоставление компьютеру шагов для выполнения Алгоритм - набор инструкций, описывающих порядок действий исполнителя для решения некоторой задачи 	2	ОПК-5.3.1																

	3. Алгоритм – это точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату.		
15.	<p>Инструкция: Проанализируйте фрагмент программного кода. В ответе запишите значение переменной c, которое будет получено в результате работы этого фрагмента (при наличии дробной части указать 2 знака после запятой):</p> <pre>int a = 10, b = 25, c = 0; while (a!=b){ if (a>b) a=a-b; else b=b-a; } c=a;</pre>	5	ОПК-5.У.1, ОПК-5.В.1
16.	<p>Инструкция: Проанализируйте фрагмент программного кода. В ответе запишите значение переменной c, которое будет получено в результате работы этого фрагмента (при наличии дробной части указать 2 знака после запятой):</p> <pre>int a = 2, b = 2, c = 0; if (a<b) c=b-a; else if (a>b) c = a-b;</pre>	0	ОПК-5.У.1, ОПК-5.В.1
17.	<p>Инструкция: Проанализируйте фрагмент программного кода. В ответе запишите значение переменной x, которое будет получено в результате работы этого фрагмента (при наличии дробной части указать 2 знака после запятой):</p> <p>Найдите значения переменной x:</p> <pre>int x; x = int(3.8)+3.3;</pre>	6	ОПК-5.У.1, ОПК-5.В.1
18.	<p>Инструкция: Проанализируйте фрагмент программного кода. В ответе запишите значение переменной x, которое будет получено в результате работы этого фрагмента (при наличии дробной части указать 2 знака после запятой):</p> <pre>int x; x = 22.0*3/5;</pre>	13	ОПК-5.У.1, ОПК-5.В.1
19.	<p>Инструкция: Прочитайте текст, выберите правильные варианты ответа</p> <p>К свойствам алгоритма в программировании относят:</p> <p>1. Однозначность</p>	1,3,4,6	ОПК-5.3.1

	<p>2. Аддитивность</p> <p>3. Конечность</p> <p>4. Результативность</p> <p>5. Индифферентность</p> <p>6. Универсальность</p>																		
20.	<p>Инструкция: Для каждого типа данных, указанных в левом столбце, подберите соответствующую область допустимых значений, указанную в правом столбце</p> <table border="1"> <tr> <td>A</td> <td>Число</td> <td>1</td> <td>от -32767 до 32768</td> </tr> <tr> <td>B</td> <td>Символ</td> <td>2</td> <td>"истина", "ложь"</td> </tr> <tr> <td>C</td> <td>Указатель</td> <td>3</td> <td>от 000000 до 999999</td> </tr> <tr> <td>D</td> <td>Логическое данное</td> <td>4</td> <td>от ' ' до 'я'</td> </tr> </table>	A	Число	1	от -32767 до 32768	B	Символ	2	"истина", "ложь"	C	Указатель	3	от 000000 до 999999	D	Логическое данное	4	от ' ' до 'я'	A1 B4 C3 D2	ОПК-5.3.1
A	Число	1	от -32767 до 32768																
B	Символ	2	"истина", "ложь"																
C	Указатель	3	от 000000 до 999999																
D	Логическое данное	4	от ' ' до 'я'																
21.	<p>Инструкция: Прочитайте текст, выберите один правильный ответ</p> <p>Выберите корректное определение структурного программирования:</p> <p>1. Структурное программирование – это методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков.</p> <p>2. Структурное программирование – это парадигма программирования, в которой основными концепциями являются понятия объектов и классов.</p> <p>3. Структурное программирование - это парадигма программирования, при использовании которой программа или её фрагмент осмысливается как модель какого-либо формального автомата</p>	1	ОПК-5.У.1, ОПК-5.В.1																
22.	<p>Инструкция: Прочитайте текст и запишите развернутый обоснованный ответ</p> <p>Согласно структурной парадигме программирования, любую программу можно построить, используя базовые управляющие структуры. Перечислите их и дайте их краткую характеристику</p>	Существует три базовых управляющих конструкции: следование (последовательность), ветвление и цикл. Следование предполагает последовательное выполнение	ОПК-5.3.1																

		<p>команд.</p> <p>Ветвление предполагает выбор одной из нескольких возможных вариантов инструкций в зависимости от истинности или ложности некоторого условия.</p> <p>Цикл предполагает выполнения блока инструкций (тела цикла) некоторое количество раз</p>	
23.	<p>Инструкция: Прочитайте текст, выберите правильные варианты ответа</p> <p>В языке C++:</p> <ol style="list-style-type: none"> 1. После инициализации ссылке может быть присвоена другая переменная 2. Ссылка всегда должна инициализироваться при объявлении, кроме случаев, когда она является параметром функции. 3. Тип ссылки должен совпадать с типом величины, на которую она ссылается. 4. Ссылки не могут иметь значение NULL. 5. Ссылка может не быть инициализированной при объявлении 	2,3,4	ПК-6.3.1
24.	<p>Инструкция: Запишите типы данных в языке программирования C/C++ в порядке, соответствующем «увеличению» типа при неявном приведении типов:</p> <ol style="list-style-type: none"> 1. signed int 2. unsigned int 3. signed char 4. unsigned char 5. float 6. double 	3,4,1,2,5,6	ПК-6.У.1
25.	<p>Инструкция: Дайте определение формальным и фактическим параметрам функции.</p>	Формальные параметры указываются в объявлении функции,	ПК-6.3.1

		фактические параметры передаются в функцию при ее вызове.	
26.	<p>Инструкция: Прочитайте текст, выберите правильные варианты ответа</p> <p>Определите в языке программирования C/C++ тип константы 2.e-2</p> <ol style="list-style-type: none"> 1. char 2. int 3. double 4. float 5. string 6. bool 	3	ПК-6.У.1
27.	<p>Инструкция: Прочитайте текст, выберите правильные варианты ответа</p> <p>Определите в языке программирования C/C++ тип константы '\b'</p> <ol style="list-style-type: none"> 1. char 2. int 3. double 4. float 5. string 6. bool 	1	ПК-6.У.1

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 основных теоретических положений курса, освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции, основные положения лекционного

материала закрепляются выполнением лабораторных работ. Теоретические материалы изложены в рекомендованной литературе, часть теоретического материала, изучаемого на лекциях изложена в методических указаниях по выполнению лабораторных работ.

В личных кабинетах обучающихся размещаются презентации к лекциям, если они используются при изложении материала, а также необходимые учебные материалы.

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.

Требования к проведению практических занятий

Практические занятия реализуются как групповая (обсуждение решения) или индивидуальная работа над заданием (ответ у доски, выполнение задания на персональном компьютере), выданным преподавателем, в течение занятия. Методически рекомендации по прохождению практических занятий выкладываются в личный кабинет обучающегося.

11.3. Методические указания для обучающихся по выполнению лабораторных работ.

Содержание лабораторного практикума направлено на получение первичных знаний и умений в области структурного программирования на языке высокого уровня и закрепление теоретических знаний, полученных в лекционном курсе, выполнением практических заданий.

Подробные методические указания по прохождению лабораторных работ и правила оформления отчетов приведены в методических указаниях [электронный ресурс кафедры №43 в локальной сети кафедры], путь \\dcbm\Методическое обеспечение кафедры 43\ Основы программирования, доступных студентам из локальной сети учебной лаборатории и размещаемых в личных кабинетах обучающихся, а также размещаются в личных кабинетах студентов.

К каждой лабораторной работе студент выполняет задания в соответствии с номером варианта, оформляет отчет и защищает работу преподавателю. Выполнение и защита всех лабораторных работ является обязательным требованием для допуска студента к экзамену.

Отчет о лабораторной работе должен содержать следующие разделы:

- Титульный лист
- Цель работы
- Задание на лабораторную работу
- Ход работы
 - Описание разработанных функций
 - Описание алгоритмов
 - Тестовые данные
 - Листинги программного кода
 - Результаты тестирования
- Выводы
- отчет о работе оформляется в виде *.doc или *.pdf-файла. Имя файла с отчетом должно содержать № группы, фамилию студента, № работы (например, М411_Иванов_ЛР1).

Титульный лист оформляется в соответствии с шаблоном, размещенном на официальном сайте ГУАП по ссылке <https://guap.ru/regdocs/docs/uch>.

Цель работы сформулирована в методических указаниях и студенту необходимо ее продублировать в отчете.

В ход работы включаются: весь разработанный программный код, набор тестовых данных, результаты работы программы на тестовых данных. Подробные требования к содержанию каждой лабораторной работы приведены в методических указаниях.

Выводы являются обязательной частью отчета о лабораторной работе и содержат результаты ее выполнения.

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 работы на языке программирования C/C++ при разработке простой программы,
- на практике применить основные этапы разработки программ в соответствии с каскадной моделью жизненного цикла ПО,
- получить навыки самостоятельной работы над программным продуктом.

Структура пояснительной записки курсовой проекта работы и Требования к оформлению пояснительной записки курсовой работы приведены в учебном пособии «Основы программирования. Основные этапы разработки программ», авторов А. А. Ключарёв, А. В. Туманова, А. А. Фоменкова, доступном как в библиотеке ГУАП, так и по ссылке

https://lib.guap.ru/jirbis2/components/com_irbis/pdf_view/?425385

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.

Методическими материалами, направляющими самостоятельную работу обучающихся является учебно-методический материал по дисциплине, указанный в настоящей рабочей программе.

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 при изучении материала лекционных занятий и лабораторного практикума.

В семестре 2 при изучении лекционного материала текущий контроль предусматривает выполнение студентами контрольной работы, выполняемой на 13 неделе семестра. Контрольная работа охватывает материал разделов 2-5 лекционных занятий и состоит в письменном ответе на теоретический вопрос, решение задачи и разработке блок-схемы алгоритма.

Выполнение контрольной работы оценивается в соответствии с модульно-рейтинговой системой.

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».

В семестре 2 для допуска к экзамену студент должен выполнить и защитить преподавателю все лабораторные работы и написать контрольную работу. На экзамене студент должен ответить на два теоретических вопроса и решить одну задачу.

В течение второго семестра студент набирает баллы в соответствии с таблицей 20. Оценка на экзамене выставляется по рейтингово-модульной системе в соответствии с таблицами 14 и 21.

Таблица 20 – Баллы за освоение материалов курса

№	Вид деятельности	Количество баллов
1	Выполнение лабораторных работ	40
2	Ответы на практических занятиях	15
3	Выполнение контрольной работы	15
4	Посещаемость лекций	10
5	Экзамен	20
	ИТОГО	100

Таблица 21 – Соответствие оценки количеству набранных баллов

Оценка	Количество баллов
отлично	>85
хорошо	70-85
удовлетворительно	55-70
неудовлетворительно	<55

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