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

Кафедра № 31

УТВЕРЖДАЮ
Руководитель образовательной программы

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

С.В. Солёный
(инициалы, фамилия)

(подпись)
«27» июня 2024 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Алгоритмизация и программирование»
(Наименование дисциплины)

Код направления подготовки/ специальности	13.05.02
Наименование направления подготовки/ специальности	Специальные электромеханические системы
Наименование направленности	Электромеханические системы специальных устройств и изделий
Форма обучения	очная
Год приема	2024

Санкт-Петербург– 2024

Лист согласования рабочей программы дисциплины

Программу составил (а)

доцент, к.т.н	27.06.2024	О.С. Нуйя
(должность, уч. степень, звание)	(подпись, дата)	(инициалы, фамилия)

Программа одобрена на заседании кафедры № 31

«27» июня 2024 г, протокол № 8

Заведующий кафедрой № 31

д.т.н.,проф.	27.06.2024	В.Ф. Шишлаков
(уч. степень, звание)	(подпись, дата)	(инициалы, фамилия)

Заместитель директора института №3 по методической работе

ст. преподаватель	27.06.2024	Н.В. Решетникова
(должность, уч. степень, звание)	(подпись, дата)	(инициалы, фамилия)

Аннотация

Дисциплина «Алгоритмизация и программирование» входит в образовательную программу высшего образования – программу специалитета по направлению подготовки/ специальности 13.05.02 «Специальные электромеханические системы» направленности «Электромеханические системы специальных устройств и изделий». Дисциплина реализуется кафедрой «№31».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-1 «Способен осуществлять поиск, обработку и анализ информации из различных источников и представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий»

ОПК-2 «Способен соблюдать основные требования информационной безопасности, в том числе требования защиты государственной тайны»

ПК-2 «Способность участвовать в конструировании электротехнических и электроэнергетических устройств, специальных электромеханических систем»

Содержание дисциплины охватывает круг вопросов, связанных с программированием и основами алгоритмизации на языке программирования С и С++.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа студента.

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

Общая трудоемкость освоения дисциплины составляет 4 зачетных единицы, 144 часа.

Язык обучения по дисциплине «русский »

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО). Основная цель преподавания дисциплины «Алгоритмизация и программирование» заключается в изучении общих принципов прикладного программирования, изучение применения типовых структур алгоритмов для решения задач, знакомство с объектно-ориентированным и функциональным программированием, а также ознакомление студентов с возможностями применения современных вычислительных средств при практическом решении инженерных задач на основе последних достижений в области программирования и автоматизации инженерных расчетов. Цель дисциплины состоит в получении студентами необходимых теоретических и практических навыков в области программирования на языке С и С++.

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-1 Способен осуществлять поиск, обработку и анализ информации из различных источников и представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	ОПК-1.З.2 знает методы разработки алгоритмов и программного обеспечения в рамках систем искусственного интеллекта ОПК-1.У.1 умеет алгоритмизировать решение задач и реализовать алгоритмы с использованием программных средств ОПК-1.В.1 владеет навыками применения средств информационных технологий для поиска, хранения, обработки, анализа и представления информации, в том числе с использованием технологий искусственного интеллекта
Общепрофессиональные компетенции	ОПК-2 Способен соблюдать основные требования информационной безопасности, в том числе требования защиты государственной тайны	ОПК-2.З.1 знает принципы работы современных информационных сетей; виды информационных и образовательных технологий, основные требования информационной безопасности, в том числе требования защиты государственной тайны
Профессиональные компетенции	ПК-2 Способность участвовать в конструировании электротехнических и электроэнергетических устройств,	ПК-2.З.2 использует методы разработки оригинальных алгоритмов и программных решений с использованием современных технологий

	специальных электромеханических систем	

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 используются при изучении других дисциплин:

− «Системное программное обеспечение».

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
1	2	3
Общая трудоемкость дисциплины, ЗЕ/ (час)	4/ 144	4/ 144
Из них часов практической подготовки	11	11
Аудиторные занятия, всего час.	51	51
в том числе:		
лекции (Л), (час)	17	17
практические/семинарские занятия (ПЗ), (час)		
лабораторные работы (ЛР), (час)	34	34
курсовой проект (работа) (КП, КР), (час)		
экзамен, (час)	27	27
Самостоятельная работа, всего (час)	66	66
***Вид промежуточной аттестации*:** зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	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)	ЛР (час)	КП (час)	СРС (час)
Семестр 2					
Раздел 1. Структура персональной ЭВМ Тема 1.1. Размещение данных и программ в памяти ЭВМ Тема 1.2. Программные модули. Ошибки Тема 1.3. Функциональная и модульная	 0.2 0.2		 3		5

декомпозиция. Тема 1.4. Файловая система хранения информации Тема 1.5. Операционная система	0.2 0.2 0.2				
Раздел 2. Основы алгоритмизации Тема 2.1. Основные свойства алгоритма Тема 2.2. Общие принципы разработки алгоритмов. Примеры алгоритмизации задач	1		2		8
Раздел 3. Языки программирования и основные понятия алгоритмического языка Тема 3.1. Алгоритм, язык программирования, программа Тема 3.2 Компиляторы и интерпретаторы Тема 3.3 Уровни языков программирования Тема 3.4 Состав и описание алгоритмического языка	1 1		2		8
Раздел 4. Введение в язык программирования Си Тема 4.1. Алфавит языка Си. Элементарные конструкции (лексемы) языка Си Тема 4.2. Концепция типа данных. Типы данных. Структура программы Тема 4.3. Операции и выражения. Алгоритм и операторы	0.5 0.5 1		2 2 1		8
Раздел 5. Операторы простой последовательности действий Тема 5.1. Функции форматного ввода-вывода данных. Функции ввода-вывода символов. Ввод-вывод данных в языке С++. Основные библиотечные функции Тема 5.2. Примеры задач на использование операторов простой последовательности и библиотечных функций	1 1 1		2 3		8
Раздел 6. Условные конструкции: операторы ветвления Тема 6.1. Условный оператор. Примеры задач на использование условного оператора Тема 6.2. Оператор множественного выбора (переключатель). Пример задачи на использование оператора множественного выбора	1 1		4		8
Раздел 7. Условные конструкции: операторы циклов Тема 7.1. Оператор цикла с параметром (счетчиком). Пример задачи на использование оператора цикла с параметром Тема 7.2. Итерационные циклы. Примеры задач на использование итерационных циклов (с предусловием и с постусловием)	1 1		3		8

Раздел 8. Указатели и массивы данных Тема 8.1. Указатели Тема 8.2. Массив как статическая структура данных и адресная арифметика Тема 8.3. Линейный поиск и сортировка в массивах данных Тема 8.4. Рекомендации при работе со статическим массивом данных	1 1 1		4 2		8
Раздел 9.Строки Тема 9.1. Символьные и строковые литералы и переменные Тема 9.2. Операции со строками	1		4		5
Итого в семестре:	17		34		66
Итого	17	0	34	0	66

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	1.1. Размещение данных и программ в памяти ЭВМ: центральный процессор, оперативное запоминающее устройство, постоянное запоминающее устройство, дисковод. 1.2. Размещение данных и программ в памяти ЭВМ: программа, команда, ячейка, программа в машинных кодах 1.3. Программные модули. Ошибки. Транслятор, компилятор, интерпретатор, объектный модуль. Синтактические ошибки, логические ошибки. 1.4. Файловая система хранения информации: функциональная декомпозиция, модельная декомпозиция, файловая система, каталог, подкаталог, маршрут файла. 1.5. Операционная система: примеры операционных систем.
2	2.1. Основные свойства алгоритма: свойства алгоритмов, графическое описание алгоритмов, словесное описание алгоритмов, основные символы схемы алгоритмов. 2.2. Общие принципы разработки алгоритмов. Примеры алгоритмизации задач. Алгоритмические структуры: альтернативный, циклический, итерационный цикл, цикл с параметром. Словесный способ, графический способ, псевдокод. Структурная схема, блок-схема. Виды графических символов для построения блок-схемы алгоритма.
3	3.1. Алгоритм, язык программирования, программа: транслятор, тестирование, отладка, синтаксис языка.

	3.2. Компиляторы и интерпретаторы: основы. 3.3. Уровни языков программирования: языки программирование низкого уровня, языки программирования высокого уровня 3.4. Состав и описание алгоритмического языка: алгоритмический язык, описание языка (выражения, операторы, символы), синтактические определения,
4	4.1.Алфавит языка Си. Элементарные конструкции (лексемы) языка Си: идентификатор, служебные слова, константы, знаки операций, комментарии, разделители. 4.2. Концепция типа данных. Типы данных. Структура программы. Базовые, целочисленные, вещественные типы данных, представление символьных данных, тип void. 4.3. Операции и выражения. Алгоритм и операторы Знаки операций, операции присваивания, арифметические операции, операции отношения и логические операции, условная операция, операция явного преобразования типа.
5	5.1. Функции форматного ввода-вывода данных. Функции ввода-вывода символов. Ввод-вывод данных в языке С++. Основные библиотечные функции Ввод-вывод данных, функция форматного ввода данных, спецификаторы преобразования, символы преобразования, символы, не являющиеся разделителями, функция форматного ввода данных. Пример организации форматного ввода-вывода данных. 5.2. Примеры задач на использование операторов простой последовательности и библиотечных функций: реализация программы.
6	6.1. Условный оператор. Примеры задач на использование условного оператора Постановка задачи. Математическая модель и описательный алгоритм решения. Блок-схема алгоритма Текст программы – реализация алгоритма на языке Си. 6.2. Оператор множественного выбора (переключатель). Пример задачи на использование оператора множественного выбора Постановка задачи. Математическая модель и описательный алгоритм решения. Блок-схема алгоритма Текст программы – реализация алгоритма на языке Си.
7	7.1. Оператор цикла с параметром (счетчиком). Пример задачи на использование оператора цикла с параметром Постановка задачи. Математическая модель и описательный алгоритм решения. Блок-схема алгоритма Текст программы – реализация алгоритма на языке Си. 7.2. Итерационные циклы. Примеры задач на использование итерационных циклов (с предусловием и с постусловием) Постановка задачи. Математическая модель и описательный алгоритм решения. Блок-схема алгоритма Текст программы – реализация алгоритма на языке Си.
8	8.1. Указатели 8.2. Массив как статическая структура данных и адресная

	арифметика 8.3. Линейный поиск и сортировка в массивах данных 8.4. Рекомендации при работе со статическим массивом данных. Примеры программ
9	9.1. Символьные и строковые литералы и переменные 9.2. Операции со строками

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
Учебным планом не предусмотрено					
Всего					

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	Алгоритмы линейной структуры и их программирование на Си и Си++	4	2	1-3
2	Операторы повторения на языке Си	4	2	1-3
3	Указатели на языке Си	4	1	1-4
4	Условный оператор на Си++	4	1	4-5
5	Оператор выбора на Си++	4	1	4-5
6	Стандартные алгоритмы работы с одномерными массивами на Си++	5	1	6-8
7	Двумерный массив на Си++	5	2	6-8
8	Строки в С++	4	1	9
Всего		34	11	

4.5. Курсовое проектирование/ выполнение курсовой работы

Учебным планом не предусмотрено

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
1	2	3
Изучение теоретического материала дисциплины (ТО)	20	20
Домашнее задание (ДЗ)	10	10
Подготовка к текущему контролю успеваемости (ТКУ)	16	16
Подготовка к промежуточной аттестации (ПА)	20	20
Всего:	66	66

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)
URL: https://new.znanium.com/catalog/product/1072040	Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2020 — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-16-108363-5.	
URL: https://new.znanium.com/catalog/product/553143	Задачник-практикум по основам программирования: учебное пособие / Амелина Н.И., Невская Е.С., Русанова Я.М. - Ростов-на-Дону:Издательство ЮФУ, 2009 - 192 с.ISBN 978-5-9275-0704-7.	
URL: https://new.znanium.com/catalog/product/1054007	Колдаев, В. Д. Структуры и алгоритмы обработки данных: Учебное пособие / Колдаев В.Д. - М.:ИЦ РИОР, НИЦ ИНФРА-М, 2020 - 296 с.: - (Высшее образование: Бакалавриат) — www.dx.doi.org/10.12737/2833. - ISBN 978-5-16-101275-8. - Текст : электронный.	
URL: https://new.znanium.com/cat	Абрамян, М. Э. Введение в стандартную библиотеку шаблонов С++. Описание, примеры использования,	

alog/product/1020515	учебные задачи : учебник / М. Э. Абрамян ; Южный федеральный университет. — Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета. 2017 — 178 с. - ISBN 978-5-9275-2374-0. - Текст : электронный.	
URL: https://new.znanium.com/catalog/product/550811	Русанова, Я. М. C++ как второй язык в обучении приемам и технологиям программирования: учеб. пособие / Я. М. Русанова. - Ростов-на-Дону: Издательство ЮФУ, 2010 - 200 с. - ISBN 978-5-9275-0749-8. - Текст : электронный.	
URL: https://new.znanium.com/catalog/product/940363	Культин, Н. Б. C/C++ в задачах и примерах / Н. Б. Культин. — Санкт-Петербург : БХВ-Петербург, 2015 — 285 с. - ISBN 978-5-9775-3322-5. - Текст : электронный.	

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
Видеокурс	Евгений Линский -лекции

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
	Не предусмотрено

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	Компьютерный класс (лучше с Интернетом)	

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средствдля 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; Экзаменационные билеты; Задачи; Тесты.

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»	– 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»	– 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»	– 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.

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«неудовлетворительно» «не зачтено»	– обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений.

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
1	Структура персональной ЭВМ	ОПК-1.З.2
2	Размещение данных и программ в памяти ЭВМ	ОПК-1.У.1
3	Программные модули	ОПК-1.В.1
4	Ошибки	ОПК-2.З.1
5	Функциональная и модульная декомпозиции	ПК-2.З.2
6	Файловая система хранения информации	ОПК-1.З.2
7	Операционная система	ОПК-1.У.1
8	Основные свойства алгоритма	ОПК-1.В.1
9	Общие принципы разработки алгоритмов	ОПК-2.З.1
10	Примеры алгоритмизации задач	ПК-2.З.2
11	Алгоритм, язык программирования, программа	ОПК-1.З.2
12	Компиляторы и интерпретаторы	ОПК-1.У.1
13	Уровни языков программирования	ОПК-1.В.1
14	Состав и описание алгоритмического языка	ОПК-2.З.1
15	Алфавит языка Си. Элементарные конструкции (лексемы) языка Си	ПК-2.З.2
16	Концепция типа данных	ОПК-1.З.2
17	Типы данных	ОПК-1.У.1
18	Структура программы	ОПК-1.В.1
19	Операции и выражения	ОПК-2.З.1
20	Алгоритм и операторы	ПК-2.З.2
21	Функции форматного ввода-вывода данных	ОПК-1.З.2
22	Функции ввода-вывода символов	ОПК-1.У.1
23	Ввод-вывод данных в языке С++	ОПК-1.В.1
24	Основные библиотечные функции	ОПК-2.З.1
25	Пример задачи на использование операторов простой последовательности	ПК-2.З.2
26	Пример задачи на использование операторов библиотечных функций	ОПК-1.З.2
27	Условный оператор	ОПК-1.У.1
28	Примеры задач на использование условного оператора	ОПК-1.В.1
28	Оператор множественного выбора (переключатель)	ОПК-2.З.1
30	Пример задачи на использование оператора множественного выбора	ПК-2.З.2
31	Оператор цикла с параметром (счетчиком)	ОПК-1.З.2
32	Пример задачи на использование оператора цикла с	ОПК-1.У.1

	параметром	
33	Итерационные циклы	ОПК-1.В.1
34	Примеры задач на использование итерационных циклов (с предусловием)	ОПК-2.З.1
35	Примеры задач на использование итерационных циклов (с постусловием)	ПК-2.З.2
36	Разработать алгоритм суммирования *n* введенных чисел, вычисления их среднего арифметического значения и вывода полученных значений.	ОПК-1.З.2
37	Разработать алгоритм суммирования положительных из *n* введенных чисел и вывода полученного значения суммы.	ОПК-1.У.1
38	Разработать алгоритм вычисления произведения только отрицательных из *n* введенных чисел с вычетом из него значения первого введенного числа и вывода полученного значения.	ОПК-1.В.1
39	Вычислить и вывести на экран значение функции $2 \cdot x3+x+1$, если $x \leq 0$; $y = -2 \cdot x2+3$, если $0<x<10$; 0, если $x \geq 10$.	ОПК-2.З.1
40	Ввести координаты точки *x, y*, присвоить $z=1$, если точка принадлежит окружности с введенным радиусом *R* и центром в точке с введенными координатами *a*, *b*, и присвоить $z=0$ – в противном случае; вывести значение *z*.	ПК-2.З.2
41	Вывести на экран сообщение, в какой четверти координатной плоскости находится точка с координатами *x, y*, если $x \cdot y \neq 0$.	ОПК-1.З.2
42	Вывести минимальное и максимальное значение из трех введенных чисел *x*1, *x*2, *x*3.	ОПК-1.У.1
43	Определить, является ли введенное число *n* совершенным, т.е. равным сумме всех своих делителей, не превосходящих само число; вывести соответствующее сообщение.	ОПК-1.В.1
44	Вычислить и вывести для введенного натурального числа *n* сумму $S=1+22+33+\ldots+nn$, не используя стандартную функцию возведения в степень.	ОПК-2.З.1
45	Сумма цифр трехзначного числа кратна 7, само число также делится на 7. Найти и вывести все такие числа.	ПК-2.З.2
46	Определить и вывести все числа, кратные введенным числам *a* и *b*, меньшие $a \cdot b$.	ОПК-1.З.2
47	В массиве A=(a1, a2, ..., an) все положительные элементы, стоящие перед минимальным положительным элементом, переслать в «хвост» массива.	ОПК-1.У.1
48	Дан двумерный массив целых чисел из 4 столбцов и 3 строк. Найти максимальный элемент в массиве и заменить его нулем.	ОПК-1.З.2
49	Дана строка. Удалить слова со второго по четвёртое.	ОПК-1.У.1
50	Написать программу, которая бы по введенному месяцу выдает все приходящиеся на этот месяц праздничные дни.	ОПК-1.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
	Учебным планом не предусмотрено

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.	Укажите, в каком году впервые был стандартизован язык C++. Необходимо выбрать один правильный ответ. 1) 1978 2) 1983 3) 1989 4) 1999 5) 2003 6) 2013 7) 2011 8) 1998 9) 2002 10) 2015 11) 2008 12) 2001 13) 1995 14) 1980	ОПК-1.3.2
2.	В языке программирования C++ дан массив int array[5] = { 3, 10, 7, 9, 2}.Выберете ответ, где можно обратиться к числу 7. 1) array[7]; 2) array[2]; 3) array[3];	
3.	Дан следующий код на языке программирования C++: #include <iostream> int main() { for (int i=3; i<8; i++) { std::cou return 0; } Выберете правильный результат кода. 1) 3,4,5,6,7, 2) 3,4,5,6,7 3) 3 4 5 6 7	

	4) 01234567 5) 34567	

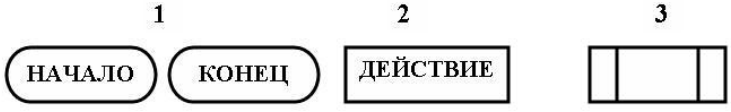
4.	Дан следующий код на языке программирования C++:
	#include <iostream>
	int main()
	{
	int x=1;
	switch(x)
	{ case 1:std::cout << "Один";
	case 0:std::cout << "Нуль";
	case 3:std::cout << "Всем привет!";
	}
	return 0;
	}
	Укажите правильный результат данного кода.
	1) ОдинНульВсем привет
	2) Один
	3) Нуль
	4) НульВсемпривет
	5) Всем привет
		ОПК-1.У.1

5.	Выберите из списка объявления, которые стоит помещать в заголовочные файлы в языке программирования C++ 1) void foo() 2) void bar() { foo(); } 3) int a; 4) void foo(); 5) extern int a;	
6.	Дан следующий код на языке программирования C++. #include <iostream> int main() { std::cout << "Доброе утро" << "/n"; std::cout <<"Сегодня есть пара?"; return 0; Укажите правильный результат данного кода.} 1) Доброе утро Сегодня есть пара? 2) Доброе утро/nСегодня есть пара? 3) Доброе утро 4) Сегодня есть пара? 5) Доброе утроСегодня есть пара?	

7.	Дан код программы на языке программирования C++:
	#include <iostream>
	int main()
	{
	int k, num=30;
	k = (num>5 ? (num <=10 ? 100 : 200): 500);
	std::cout<<num;
	return 0;
	}
	Выберете, что появится на экране (в консоли) в результате выполнения данного кода.
	1) 100	

	2) 10 3) 200 4) 500 5) 30 6) 5 7) 0	
8.	Укажите, сколько параметров можно передать в деструктор (язык программирования С++) 1) Не более 3 2) Не более 10 3) Не более 15 4) Максимум 1 5) Нельзя передавать параметры в деструктор	ОПК-1.В.1

9.	Дан код программы на языке программирования С++:
	#include <iostream>
	#include <string>
	void print(int v)
	{std::cout << "int:" << v << std::endl;}
	void print(bool v)
	{std::cout << "bool:" << v << std::endl;}
	void print(std::string v)
	{std::cout << "std::string:" << v << std::endl;}
	int main()
	{print(1);
	print(true);
	print("Hello world");
	return 0;
	}
	Выберете, что появится на экране (в консоли) в результате выполнения данного кода.
	1) int:1
	bool:1
	bool:1
	2) 1
	True
	Hello word
	3) Ничего не выведет	

10.	Задача на соответствие- основные конструкции, использующиеся для построения блок-схем. 1 2 3  а) Блок, характеризующий начало/конец алгоритма (для подпрограмм – вызов/возврат); б) Блок - процесс, предназначенный для описания отдельных действий; в) Блок - предопределенный процесс, предназначенный для обращения к вспомогательным алгоритмам (подпрограммам) 1-а) 2-б) 3-в)	

11.	Задача на соответствие. Фамилия- что изобрел. 1)Линус Торвальдс 2)Тим Бернерс-Ли 3)Джеймс Гослинг а) Linux б) создание HTTP, URL, HTML в) Создатель языка Java 1-а) 2-б) 3-в)	
12.	Задача на соответствие. Фамилия- что изобрел. 1)Брендан Эйх 2)Бьёрн Страуструп 3)Марк Цукерберг а) JavaScript б) Создатель языка С++ в) Facebook 1-а) 2-б) 3-в)	ОПК-2.3.1
13.	Задача на соответствие.Функция – команда на С++ 1)Синус 2)Косинус 3)Возведение числа в степень 4)Корень квадратный а) sin(x) б) cos(x) в) pow (x, y) г) sqrt(x) 1-а) 2-б) 3-в) 4-г)	
14.	Задача. Разработайте алгоритм, который проверяет введенное пользователем значение величины целого типа А, и если оно отрицательное, то увеличивает введенное значение на 2, иначе уменьшает его на 4, а затем выводит на экран полученное значение А. Решение. Проектирование 1 Постановка задачи. Входные данные: величина А целого типа, цел А; Выходные данные: величина А целого типа, цел А; Связь между входными и выходными данными: если А<0 то А:=А+2 иначе А:=А-4 Поставьте в правильном порядке выполняемые действия Выполняемые действия a) Начало алгоритма b) Описание используемых переменных c) Ввод данных (ввод значения переменной А, тип целый) d) Анализ введенных данных (значение переменной А отрицательное)	

	e) Присваивание значения (переменной A присваивается значение A+2) f) Присваивание значения (переменной A присваивается значение A-4) g) Вывод данных (вывод полученного значения переменной тип целый) h) Конец алгоритма 1-a) 2-b) 3-c) 4-d) 5-e) 6-f) 7-g) 8-h)	
15.	Задача. Разработайте алгоритм, который вычисляет сумму S первых n натуральных чисел и выводит на экран полученное значение S. Решение. Проектирование 1 Постановка задачи. Входные данные: величина n целого типа, цел n; Выходные данные: величина S целого типа, цел S; Связь между входными и выходными данными: для i от 1 до n шаг 1 нц S:=S+i Кц Поставьте в правильном порядке выполняемые действия Выполняемые действия a) Начало алгоритма b) Описание используемых переменных (целые n, S, i) c) Ввод данных (ввод значения переменной n, тип целый) d) Присваивание значения (переменной S присваивается значение 0) e) Анализ данных (значение переменной i от 1 до n) f) Присваивание значения (переменной S присваивается значение S + i) g) Вывод данных (вывод вычисленного значения переменной величины S, тип целый) h) Конец алгоритма 1-a) 2-b) 3-c) 4-d) 5-e) 6-f) 7-g) 8-h)	
16.	У вас есть проект на языке программирования C++, в котором есть три файла с различными определениями: utility.cpp, lexer.cpp и parser.cpp. Каждому файлу соответствует заголовочный файл с необходимыми объявлениями: utility.hpp, lexer.hpp и parser.hpp. Каждый из файлов определений непосредственно (т. е. с помощью директивы include) подключает соответствующий ему заголовочный файл с объявлениями (т. е. utility.cpp подключает utility.hpp, а lexer.cpp подключает lexer.hpp и аналогично для parser.cpp и parser.hpp). Кроме того, файл parser.hpp непосредственно подключает заголовки lexer.hpp и utility.hpp, а файл lexer.hpp подключает заголовок utility.hpp. В этой задаче вам нужно выбрать все верные утверждения из списка. Перед решением задачи полезно нарисовать дерево включений. 1) Файл utility.hpp включается только в utility.cpp	ПК-2.3.2

	2) Если в заголовке utility.hpp отсутствует защита от повторного включения, то он будет включен в parser.cpp один раз 3) Если в заголовке utility.hpp отсутствует защита от повторного включения, то он будет включен в parser.cpp дважды 4) Файл parser.cpp включает заголовочный файл utility.hpp 5) Файл utility.cpp подключает только один заголовочный файл из перечисленных	
17.	Отметьте все верные утверждения для языка программирования C++. 1) Скомпилировать программу на C++ для некоторой архитектуры X можно только на компьютере с архитектурой X. 2) Код программы, написанный на интерпретируемом языке, можно без предварительной компиляции запустить на любой операционной системе, где установлен интерпретатор этого языка. 3) Код программы, написанный на языке, который компилируется в байт код виртуальной машины, достаточно скомпилировать однажды, чтобы программу можно было запускать на любой операционной системе, где есть соответствующая виртуальная машина. 4) Для запуска программы, код которой был написан на компилируемом языке, на компьютере должен быть установлен компилятор этого языка. 5) Код программы, написанный на языке, который компилируется в машинный код, достаточно скомпилировать однажды, и потом программу можно будет запустить на любой операционной системе, для которой существует компилятор этого языка. 6) Для запуска программы, код которой был написан на интерпретируемом языке, на компьютере должен быть установлен интерпретатор этого языка.	
18.	Укажите существующие модификаторы доступа в языке программирования C++ 1) protected 2) included 3) private	
19.	Укажите, где процессор будет искать файл стандартной библиотеки ввода/вывода: #include (язык программирования C, C++) 1) Во всех папках включения файлов 2) В локальных по отношению к проекту папках включения файлов 3) В глобальных по отношению к проекту папках включения файлов	
20.	Выберете, чем является конструктор класса в языке программирования C++. 1) Метод инициализации экземпляра, который проверяет возможность создания экземпляра и реализует его 2) Способ создания класса 3) Специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями	
21.	Укажите, как правильно объявить переменную на языке	ОПК-1.3.2

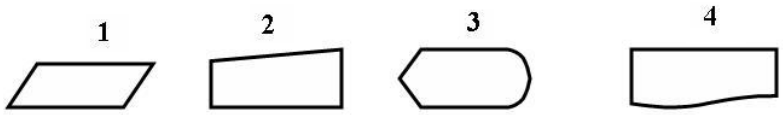
	программирования C++. 1) string text="text"; 2) char text[]='text'; 3) char text[]="text";	
22.	В языке программирования C++ дан массив int array[5] = { 3, 10, 7, 9, 2}.Выберете ответ, где можно обратиться к числу 7. 1) array[7]; 2) array[2]; 3) array[3];	
23.	Дан следующий код на языке программирования C++: #include <iostream> int main() { for (int i=3; i<8; i++) { std::cou return 0; } Выберете правильный результат кода. 1) 3,4,5,6,7, 2) 3,4,5,6,7 3) 3 4 5 6 7 4) 01234567 5) 34567	

24.	Дан следующий код на языке программирования C++:
	#include <iostream>
	int main()
	{
	int x=1;
	switch(x)
	{ case 1:std::cout << "Один";
	case 0:std::cout << "Нуль";
	case 3:std::cout << "Всем привет!";
	}
	return 0;
	}
	Укажите правильный результат данного кода.
	1) ОдинНульВсем привет
	2) Один
	3) Нуль
	4) НульВсемпривет
	5) Всем привет	

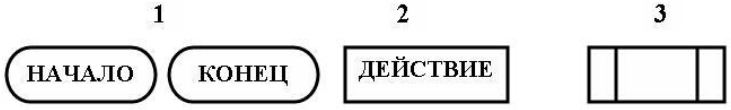
25.	Выберите из списка объявления, которые стоит помещать в заголовочные файлы в языке программирования C++ 1) void foo() 2) void bar() { foo(); } 3) int a; 4) void foo(); 5) extern int a;	ОПК-1.У.1
26.	Дан следующий код на языке программирования C++. #include <iostream> int main() { std::cout << "Доброе утро" << "/n"; std::cout <<"Сегодня есть пара?";	

	return 0; Укажите правильный результат данного кода.} 1) Доброе утро Сегодня есть пара? 2) Доброе утро/nСегодня есть пара? 3) Доброе утро 4) Сегодня есть пара? 5) Доброе утроСегодня есть пара?	

27.	Дан код программы на языке программирования C++:
	#include <iostream>
	int main()
	{
	int k, num=30;
	k = (num>5 ? (num <=10 ? 100 : 200): 500);
	std::cout<<num;
	return 0;
	}
	Выберете, что появится на экране (в консоли) в результате выполнения данного кода.
	1) 100
	2) 10
	3) 200
	4) 500
	5) 30
	6) 5
	7) 0	

28.	Задача на соответствие- основные конструкции, использующиеся для построения блок-схем.  а) Блок - ввода/вывода с неопределенного носителя; б) Блок - ввод с клавиатуры; в) Блок - вывод на монитор; г) Блок - вывод на печатающее устройство 1-а) 2-б) 3-в) 4-г)	
29.	Дан код программы на языке программирования C++: #include <iostream> #include <string> void print(int v) {std::cout << "int:" << v << std::endl;} void print(bool v) {std::cout << "bool:" << v << std::endl;} void print(std::string v) {std::cout << "std::string:" << v << std::endl;} int main() {print(1); print(true);	

	print("Hello world");
	return 0;
	}
	Выберете, что появится на экране (в консоли) в результате выполнения данного кода.
	1) int:1
	bool:1
	bool:1
	2) 1
	True
	Hello word
	3) Ничего не выведет	

30.	Задача на соответствие- основные конструкции, использующиеся для построения блок-схем.  а) Блок, характеризующий начало/конец алгоритма (для подпрограмм – вызов/возврат); б) Блок - процесс, предназначенный для описания отдельных действий; в) Блок - предопределенный процесс, предназначенный для обращения к вспомогательным алгоритмам (подпрограммам) 1-а) 2-б) 3-в)	ОПК-1.В.1
31.	Задача на соответствие. Фамилия- что изобрел. 1)Линус Торвальдс 2)Тим Бернерс-Ли 3)Джеймс Гослинг а) Linux б) создание HTTP, URL, HTML в) Создатель языка Java 1-а) 2-б) 3-в)	
32.	Задача на соответствие. Фамилия- что изобрел. 1)Брендан Эйх 2)Бьёрн Страуструп 3)Марк Цукерберг а) JavaScript б) Создатель языка С++ в) Facebook 1-а) 2-б) 3-в)	
33.	Задача на соответствие.Функция – команда на С++ 1)Синус 2)Косинус 3)Возведение числа в степень 4)Корень квадратный а) sin(x) б) cos(x)	

	в) pow (x, y) г) sqrt(x) 1-а) 2-б) 3-в) 4-г)	
34.	Задача. Разработайте алгоритм, который проверяет введенное пользователем значение величины целого типа А, и если оно отрицательное, то увеличивает введенное значение на 2, иначе уменьшает его на 4, а затем выводит на экран полученное значение А. Решение. Проектирование 1 Постановка задачи. Входные данные: величина А целого типа, цел А; Выходные данные: величина А целого типа, цел А; Связь между входными и выходными данными: если А<0 то А:=А+2 иначе А:=А-4 Поставьте в правильном порядке выполняемые действия Выполняемые действия i) Начало алгоритма j) Описание используемых переменных k) Ввод данных (ввод значения переменной А, тип целый) l) Анализ введенных данных (значение переменной А отрицательное) m) Присваивание значения (переменной А присваивается значение А+2) n) Присваивание значения (переменной А присваивается значение А-4) o) Вывод данных (вывод полученного значения переменной тип целый) p) Конец алгоритма 1-a) 2-b) 3-c) 4-d) 5-e) 6-f) 7-g) 8-h)	
35.	Задача. Разработайте алгоритм, который вычисляет сумму S первых n натуральных чисел и выводит на экран полученное значение S. Решение. Проектирование 1 Постановка задачи. Входные данные: величина n целого типа, цел n; Выходные данные: величина S целого типа, цел S; Связь между входными и выходными данными: для i от 1 до n шаг 1 нц S:=S+i Кц Поставьте в правильном порядке выполняемые действия Выполняемые действия	

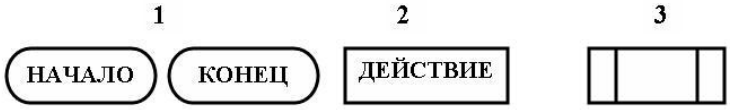
	i) Начало алгоритма j) Описание используемых переменных (целые n, S, i) k) Ввод данных (ввод значения переменной n, тип целый) l) Присваивание значения (переменной S присваивается значение 0) m) Анализ данных (значение переменной i от 1 до n) n) Присваивание значения (переменной S присваивается значение S + i) o) Вывод данных (вывод вычисленного значения переменной величины S, тип целый) p) Конец алгоритма 1-a) 2-b) 3-c) 4-d) 5-e) 6-f) 7-g) 8-h)	
36.	У вас есть проект на языке программирования C++, в котором есть три файла с различными определениями: utility.cpp, lexer.cpp и parser.cpp. Каждому файлу соответствует заголовочный файл с необходимыми объявлениями: utility.hpp, lexer.hpp и parser.hpp. Каждый из файлов определений непосредственно (т. е. с помощью директивы include) подключает соответствующий ему заголовочный файл с объявлениями (т. е. utility.cpp подключает utility.hpp, а lexer.cpp подключает lexer.hpp и аналогично для parser.cpp и parser.hpp). Кроме того, файл parser.hpp непосредственно подключает заголовки lexer.hpp и utility.hpp, а файл lexer.hpp подключает заголовок utility.hpp. В этой задаче вам нужно выбрать все верные утверждения из списка. Перед решением задачи полезно нарисовать дерево включений. 1) Файл utility.hpp включается только в utility.cpp 2) Если в заголовке utility.hpp отсутствует защита от повторного включения, то он будет включен в parser.cpp один раз 3) Если в заголовке utility.hpp отсутствует защита от повторного включения, то он будет включен в parser.cpp дважды 4) Файл parser.cpp включает заголовочный файл utility.hpp 5) Файл utility.cpp подключает только один заголовочный файл из перечисленных	ПК-2.3.2
37.	Отметьте все верные утверждения для языка программирования C++. 1) Скомпилировать программу на C++ для некоторой архитектуры X можно только на компьютере с архитектурой X. 2) Код программы, написанный на интерпретируемом языке, можно без предварительной компиляции запустить на любой операционной системе, где установлен интерпретатор этого языка. 3) Код программы, написанный на языке, который компилируется в байт код виртуальной машины, достаточно скомпилировать однажды, чтобы программу можно было запускать на любой операционной системе, где есть соответствующая виртуальная машина. 4) Для запуска программы, код которой был написан на компилируемом языке, на компьютере должен быть установлен компилятор этого языка. 5) Код программы, написанный на языке, который компилируется в машинный код, достаточно скомпилировать однажды, и потом	

	программу можно будет запустить на любой операционной системе, для которой существует компилятор этого языка. 6) Для запуска программы, код которой был написан на интерпретируемом языке, на компьютере должен быть установлен интерпретатор этого языка.	
38.	Укажите существующие модификаторы доступа в языке программирования C++ 1) protected 2) included 3) private	
39.	Укажите, где процессор будет искать файл стандартной библиотеки ввода/вывода: #include (язык программирования C, C++) 1) Во всех папках включения файлов 2) В локальных по отношению к проекту папках включения файлов 3) В глобальных по отношению к проекту папках включения файлов	
40.	Выберете, чем является конструктор класса в языке программирования C++. 1) Метод инициализации экземпляра, который проверяет возможность создания экземпляра и реализует его 2) Способ создания класса 3) Специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями	
№ п/п	Примерный перечень вопросов для тестов	Код индикатора
41.	Укажите, где процессор будет искать файл стандартной библиотеки ввода/вывода: #include (язык программирования C, C++) 1) Во всех папках включения файлов 2) В локальных по отношению к проекту папках включения файлов 3) В глобальных по отношению к проекту папках включения файлов	ОПК-2.3.1
42.	Выберете, чем является конструктор класса в языке программирования C++. 1) Метод инициализации экземпляра, который проверяет возможность создания экземпляра и реализует его 2) Способ создания класса 3) Специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями	
43.	Выберете, как правильно объявить одномерный массив с именем array, состоящий из 10 элементов целочисленного типа в языке программирования C++. 1) int array[10]; 2) array[10]; 3) int array(10);34567	
	В языке программирования C++ дан массив int array[5] = { 3, 10, 7, 9, 2}.Выберете ответ, где можно обратиться к числу 7.	

	1) array[7]; 2) array[2]; 3) array[3];	
44.	Выберите из списка объявления, которые стоит помещать в заголовочные файлы в языке программирования C++ 1) void foo() 2) void bar() { foo(); } 3) int a; 4) void foo(); 5) extern int a;	
45.	Дан следующий код на языке программирования C++. #include <iostream> int main() { std::cout << "Доброе утро" << "/n"; std::cout <<"Сегодня есть пара?"; return 0; Укажите правильный результат данного кода.} 1) Доброе утро Сегодня есть пара? 2) Доброе утро/nСегодня есть пара? 3) Доброе утро 4) Сегодня есть пара? 5) Доброе утроСегодня есть пара?	

46.	Дан код программы на языке программирования C++:
	#include <iostream>
	int main()
	{
	int i=4;
	int a=12;
	while(i<5)
	{a =+ i;
	i++;
	std::cout << a;
	}
	return 0;
	}
	Укажите значение переменной «a» после выполнения этого цикла?
	1) 4
	2) 5
	3) 0
	4) 12
		ОПК-1.В.1

47.	Дан код программы на языке программирования C++:
	#include <iostream>
	int x=1;
	int main()
	{ int g=90;
	g +=2;
	std::cout << g;
	return 0;
	}
	Выберете, что появится на экране (в консоли) в результате	

	выполнения данного кода. 1) 2 2) 90 3) g 4) 92 5) x 6) 1	
48.	Укажите, сколько параметров можно передать в деструктор (язык программирования C++) 1) Не более 3 2) Не более 10 3) Не более 15 4) Максимум 1 5) Нельзя передавать параметры в деструктор	
49.	Задача на соответствие- основные конструкции, использующиеся для построения блок-схем.  а) Блок, характеризующий начало/конец алгоритма (для подпрограмм – вызов/возврат); б) Блок - процесс, предназначенный для описания отдельных действий; в) Блок - предопределенный процесс, предназначенный для обращения к вспомогательным алгоритмам (подпрограммам) 1-а) 2-б) 3-в)	
50.	Задача на соответствие. Фамилия- что изобрел. 1)Линус Торвальдс 2)Тим Бернерс-Ли 3)Джеймс Гослинг а) Linux б) создание HTTP, URL, HTML в) Создатель языка Java 1-а) 2-б) 3-в)	
51.	Задача на соответствие. Фамилия- что изобрел. 1)Брендан Эйх 2)Бьёрн Страуструп 3)Марк Цукерберг а) JavaScript б) Создатель языка C++ в) Facebook 1-а) 2-б) 3-в)	
52.	Задача. Разработайте алгоритм, который проверяет введенное пользователем значение величины целого типа A, и если оно отрицательное, то увеличивает введенное значение на 2, а затем выводит на экран полученное значение A. Решение	ПК-2.З.2

	Проектирование 1 Постановка задачи: входные данные: величина A целого типа, цел A; выходные данные: величина A целого типа, цел A; связь между входными и выходными данными: если A<0 то A:=A+2. Поставьте в правильном порядке выполняемые действия Выполняемые действия а) Начало алгоритма б) Описание используемых переменных в) Ввод данных (ввод значения переменной A, тип целый) г) Анализ введенных данных (значение переменной A отрицательное) д) Присваивание значения (переменной A присваивается значение A+2) е) Вывод данных (полученное значение переменной A, тип целый) з) Конец алгоритма 1-а) 2-б) 3-в) 4-г) 5-д) 6-е) 7-з)	
53.	Задача. Разработайте алгоритм, который вычисляет сумму S первых n натуральных чисел и выводит на экран полученное значение S. Проектирование: 1 Постановка задачи: входные данные: величина n целого типа, цел n; выходные данные: величина S целого типа, цел S; связь между входными и выходными данными: пока i <= n. нц S := S + i; i:=i+1 кц Поставьте в правильном порядке выполняемые действия Выполняемые действия a) Начало алгоритма b) Описание используемых переменных (целые n, S, i) c) Ввод данных (ввод значения переменной n, тип целый) d) Присваивание значения (переменной S присваивается значение 0) e) Присваивание значения (переменной i присваивается значение 1) f) Анализ данных (значение переменной i не превосходит n) g) Присваивание значения (переменной S присваивается значение S + i) h) Присваивание значения (переменной i присваивается значение i+1) i) Вывод данных (вывод вычисленного значения переменной S, тип целый) j) Конец алгоритма 1-a) 2-b) 3-c) 4-d) 5-e) 6-f) 7-g) 8-h) 9-i) 10-j)	

54.	Задача. Разработайте алгоритм, который вычисляет сумму S первых n натуральных чисел и выводит на экран полученное значение S. Решение. Проектирование 1 Постановка задачи. Входные данные: величина n целого типа, цел n; Выходные данные: величина S целого типа, цел S; Связь между входными и выходными данными: выполнять S:= S+i i:=i+1 до i>n Поставьте в правильном порядке выполняемые действия Выполняемые действия a) Начало алгоритма b) Описание используемых переменных (целые n, S, i) c) Ввод данных (ввод значения переменной n, тип целый) d) Присваивание значения (переменной S присваивается значение 0) e) Присваивание значения (переменной i присваивается значение 1) f) Присваивание значения (переменной S присваивается значение S+i) g) Присваивание значения (переменной i присваивается значение i+1) h) Анализ данных (значение переменной i превосходит n) i) Вывод данных (вывод полученного значения переменной S, тип целый) j) Конец алгоритма 1-a) 2-b) 3-c) 4-d) 5-e) 6-f) 7-g) 8-h) 9-i) 10-j)	
55.	Задача. Разработайте алгоритм, который вычисляет сумму S первых n натуральных чисел и выводит на экран полученное значение S. Решение. Проектирование 1 Постановка задачи. Входные данные: величина n целого типа, цел n; Выходные данные: величина S целого типа, цел S; Связь между входными и выходными данными: для i от 1 до n шаг 1 нц S:=S+i Кц Поставьте в правильном порядке выполняемые действия Выполняемые действия q) Начало алгоритма r) Описание используемых переменных (целые n, S, i)	

	s) Ввод данных (ввод значения переменной n, тип целый) t) Присваивание значения (переменной S присваивается значение 0) u) Анализ данных (значение переменной i от 1 до n) v) Присваивание значения (переменной S присваивается значение S + i) w) Вывод данных (вывод вычисленного значения переменной величины S, тип целый) x) Конец алгоритма 1-a) 2-b) 3-c) 4-d) 5-e) 6-f) 7-g) 8-h)	
56.	У вас есть проект на языке программирования C++, в котором есть три файла с различными определениями: utility.cpp, lexer.cpp и parser.cpp. Каждому файлу соответствует заголовочный файл с необходимыми объявлениями: utility.hpp, lexer.hpp и parser.hpp. Каждый из файлов определений непосредственно (т. е. с помощью директивы include) подключает соответствующий ему заголовочный файл с объявлениями (т. е. utility.cpp подключает utility.hpp, а lexer.cpp подключает lexer.hpp и аналогично для parser.cpp и parser.hpp). Кроме того, файл parser.hpp непосредственно подключает заголовки lexer.hpp и utility.hpp, а файл lexer.hpp подключает заголовок utility.hpp. В этой задаче вам нужно выбрать все верные утверждения из списка. Перед решением задачи полезно нарисовать дерево включений. 1) Файл utility.hpp включается только в utility.cpp 2) Если в заголовке utility.hpp отсутствует защита от повторного включения, то он будет включен в parser.cpp один раз 3) Если в заголовке utility.hpp отсутствует защита от повторного включения, то он будет включен в parser.cpp дважды 4) Файл parser.cpp включает заголовочный файл utility.hpp 5) Файл utility.cpp подключает только один заголовочный файл из перечисленных	
57.	Отметьте все верные утверждения для языка программирования C++. 1) Скомпилировать программу на C++ для некоторой архитектуры X можно только на компьютере с архитектурой X. 2) Код программы, написанный на интерпретируемом языке, можно без предварительной компиляции запустить на любой операционной системе, где установлен интерпретатор этого языка. 3) Код программы, написанный на языке, который компилируется в байт код виртуальной машины, достаточно скомпилировать однажды, чтобы программу можно было запускать на любой операционной системе, где есть соответствующая виртуальная машина. 4) Для запуска программы, код которой был написан на компилируемом языке, на компьютере должен быть установлен компилятор этого языка. 5) Код программы, написанный на языке, который компилируется в машинный код, достаточно скомпилировать однажды, и потом программу можно будет запустить на любой операционной системе, для которой существует компилятор этого языка.	

	6) Для запуска программы, код которой был написан на интерпретируемом языке, на компьютере должен быть установлен интерпретатор этого языка.	
58.	Укажите существующие модификаторы доступа в языке программирования C++ 1) protected 2) included 3) private	
59.	Укажите, где процессор будет искать файл стандартной библиотеки ввода/вывода: #include (язык программирования C, C++) 1) Во всех папках включения файлов 2) В локальных по отношению к проекту папках включения файлов 3) В глобальных по отношению к проекту папках включения файлов	
60.	Выберете, чем является конструктор класса в языке программирования C++. 1) Метод инициализации экземпляра, который проверяет возможность создания экземпляра и реализует его 2) Способ создания класса 3) Специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями	

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.

Структура предоставления лекционного материала:

Структура персональной ЭВМ;

Основы алгоритмизации;

Языки программирования и основные понятия алгоритмического языка;

Введение в язык программирования Си++;

Операторы простой последовательности действий;

Условные конструкции: операторы ветвления;

Условные конструкции: операторы циклов.

Лекционный материал имеется в виде файлов.

11.2. 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

Студентам выдается индивидуальное задание для составления программы.

Работоспособность программы проверяется преподавателем.

Структура и форма отчета о лабораторной работе

Отчет по каждой лабораторной работе должен содержать цель работы, формулировку задания, текст программы с комментариями и контрольный или контрольные примеры, подтверждающие правильность работы программы.

Требования к оформлению отчета о лабораторной работе

Отчет должен содержать титульный лист, а его содержание должно быть оформлено согласно ГОСТ 7.32 – 2017 Нормативная документация, необходимая для оформления, приведена на электронном ресурсе ГУАП: https://guap.ru/standart/doc

11.3. 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.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

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 обучающихсяявляются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

11.4. 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.

11.5. 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».

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