

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

Кафедра № 2

УТВЕРЖДАЮ

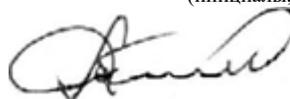
Руководитель направления

проф., д. пед. н., доц.

(должность, уч. степень, звание)

А.Г. Степанов

(инициалы, фамилия)



(подпись)

« 22 » _____ 06 _____ 2023 __ г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы программирования»
(Наименование дисциплины)

Код направления подготовки/ специальности	09.03.03
Наименование направления подготовки/ специальности	Прикладная информатика
Наименование направленности	Цифровая инфраструктура обеспечивающих систем
Форма обучения	очная

Санкт-Петербург– 2023 __

Лист согласования рабочей программы дисциплины

Программу составил (а)

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



(подпись, дата)

В.А.Галанина

(инициалы, фамилия)

Программа одобрена на заседании кафедры № 2

« 22 » 06 2023 г, протокол № 12/22-23

Заведующий кафедрой № 2

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



(подпись, дата)

В.Г. Фарафонов

(инициалы, фамилия)

Ответственный за ОП ВО 09.03.03(04)

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



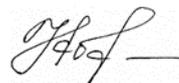
(подпись, дата)

В.А. Галанина

(инициалы, фамилия)

Заместитель директора института ФПТИ по методической работе

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



(подпись, дата)

Ю.А. Новикова

(инициалы, фамилия)

Аннотация

Дисциплина «Основы программирования» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.03 «Прикладная информатика» направленности «Цифровая инфраструктура обеспечивающих систем». Дисциплина реализуется кафедрой «№2».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

УК-2 «Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений»

ОПК-2 «Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности»

ОПК-7 «Способен разрабатывать алгоритмы и программы, пригодные для практического применения»

ПК-3 «Способен анализировать возможность реализации требований к программному обеспечению обеспечивающих систем»

ПК-6 «Способен разрабатывать базы данных обеспечивающих систем»

ПК-8 «Способен руководить разработкой программного кода»

Содержание дисциплины охватывает круг вопросов, связанных с изучением принципов алгоритмизации решения прикладных задач и реализации этих алгоритмов с использованием языка программирования высокого уровня.

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

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 обучающимися необходимых знаний по теории алгоритмов, практических навыков алгоритмизации вычислительных задач, умения реализовывать алгоритмы с использованием языка высокого уровня (на примере языка Си), знания основ объектно-ориентированного программирования (на примере языка C++)

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
Универсальные компетенции	УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.В.3 владеть навыками использования цифровых средств для решения поставленной задачи
Общепрофессиональные компетенции	ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.3.1 знать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности ОПК-2.У.1 уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности ОПК-2.В.1 владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
Общепрофессиональные компетенции	ОПК-7 Способен разрабатывать алгоритмы и программы,	ОПК-7.3.1 знать основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные

	пригодные для практического применения	среды разработки информационных систем и технологий ОПК-7.У.1 уметь применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ ОПК-7.В.1 владеть навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач
Профессиональные компетенции	ПК-3 Способен анализировать возможность реализации требований к программному обеспечению обеспечивающих систем	ПК-3.3.3 знать методологию разработки программного обеспечения и технологию программирования ПК-3.У.3 уметь применять существующие стандарты для разработки технической документации на компьютерное программное обеспечение
Профессиональные компетенции	ПК-6 Способен разрабатывать базы данных обеспечивающих систем	ПК-6.3.5 знать основы программирования ПК-6.3.6 знать современные объектно-ориентированные языки программирования ПК-6.3.7 знать современные структурные языки программирования ПК-6.В.1 владеть современными объектно-ориентированными языками программирования
Профессиональные компетенции	ПК-8 Способен руководить разработкой программного кода	ПК-8.3.1 знать методы и приемы алгоритмизации поставленных задач ПК-8.У.1 уметь писать программный код на выбранном языке программирования

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:

– «Вычислительная математика»,

– «Технология программирования»,

– «Программная инженерия»

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	59	34	25
Аудиторные занятия, всего час.	187	102	85
в том числе:			
лекции (Л), (час)	68	34	34
практические/семинарские занятия (ПЗ), (час)	34	34	
лабораторные работы (ЛР), (час)	68	34	34
курсовой проект (работа) (КП, КР), (час)	17		17
экзамен, (час)	90	54	36
Самостоятельная работа, всего (час)	47	24	23
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз., Экз.	Экз.	Экз.

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. Основы алгоритмизации и программирования					
Тема 1.1. Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. ГОСТ на изображение алгоритмов	1	8			
Тема 1.2. Базовые и дополнительные управляющие структуры	2	10			
Тема 1.3. Реализация базовых и дополнительных управляющих структур на алгоритмических языках.	2	16			
Раздел 2. Основы языка Си.					
Тема 2.1. История развития языка Си. Стандарты языка.	1				
Тема 2.2. Типы данных, операции, выражения					

Тема 2.3. Реализация потоков управления	4				
Тема 2.4. Функции и структура программы на Си	4		6		
Тема 2.5 Указатели и массивы	4				
Тема 2.6. Структуры	4		8		
Тема 2.7. Форматный ввод/вывод	4		8		
Тема 2.8. Работа с файлами	4		4		
	4		8		
Итого в семестре:	34	34	34		24
Семестр 3					
Раздел 3. Основы объектно-ориентированного программирования					
Тема 3.1. Объекты и классы.	5		8		
Тема 3.2. Наследование классов	4		10		
Тема 3.3. Классы и динамическое выделение памяти	4		8		
Тема 3.4. Контейнерные классы	4		8		
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
Раздел 1.	Основы алгоритмизации и программирования. Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. Общие принципы построения алгоритмов. Базовые управляющие структуры: линейный процесс, ветвящийся процесс (развилка), циклический процесс (цикл с предусловием). Дополнительные управляющие структуры: обход, выбор, цикл с постусловием, цикл с параметром. Принципы структурного программирования.
Раздел 2.	Основы языка Си. История языка Си. Стандарты языка. Этапы решения задачи на ЭВМ. Структура программы на языке Си. Базовые типы данных. Операции, выражения,

	<p>операторы. Преобразования типов. . Управляющие операторы: операторы while, for, do while. Операторы ветвления: if...else, оператор выбора switch. Использование функций. Аргументы функций. Передача по значению и использование указателей. Создание прототипов функций. Рекурсия. Локальные и глобальные переменные. Время жизни и область видимости. Массивы и указатели. Указатели и многомерные массивы. Динамические массивы. Функция malloc(). Файловый ввод/вывод. Обработка символьной и строковой информации. Структуры. Инициализация структур. Указатели на структуры. Перечислимые типы. Понятие пространства имен. Препроцессор и библиотека Си.</p>
Раздел 3.	<p>Основы объектно-ориентированного программирования. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. История возникновения и развития C++. Специальные типы данных. Дополнительные сведения о функциях. Модели памяти и пространство имен. Объекты и классы. Массивы объектов. Перегрузка операций. Дружественные функции. Динамическая память и классы. Наследование классов: открытое, закрытое и защищенное. Управление доступом. Абстрактные базовые классы. Виртуальные функции и классы. Шаблоны функций и классов. Контейнерные классы. Новый стандарт C++.</p>

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)	№ раздела дисциплины
Семестр 2				
1	Составление блок-схем линейных алгоритмов	Выполнение практического задания в соответствии с заданием; Наблюдение за деятельностью обучающихся на практических занятиях.	8	1
2	Составление блок-схем ветвящихся алгоритмов		8	1
3	Составление блок-схем циклических алгоритмов		10	1
4	Составление блок-схем алгоритмов сортировки данных		8	1
Всего			34	

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	№ раздела дисциплины
Семестр 2			
1	Структурное программирование. Язык Си. Поиск экстремальных элементов	2	2
2	Программирование ветвящихся процессов. Решение геометрических задач.	4	2
3	Обработка числовой последовательности	4	2
4	Организация циклов в языке Си.	4	2
5	Работа со статическими массивами в языке Си	4	2
6	Использование указателей в языке Си. Обработка двумерных массивов.	4	2
7	Использование функций в языке Си.	4	2
8	Работа со сложными типами данных в Си.	4	2
9	Обработка строковых переменных. Класс string (введение в язык C++).	4	2
Семестр 3			
1	Динамические массивы. Программирование задач линейной алгебры.	4	3
2	Работа с составными типами данных. Структуры	2	3
3	Работа с текстовыми файлами. Класс fstream.	4	3
4	Разработка базы данных средствами C++	4	3
5	Классы в C++. Работа с графической библиотекой S	4	3
6	Работа с классами. Наследование	4	3
7	Работа с классами. Абстрактные базовые классы.	4	3
8	Обработка структур данных. Стек	4	3
9	Обработка структур данных. Очередь	4	3
Всего		68	

4.5. Курсовое проектирование/ выполнение курсовой работы

Цель курсовой работы:

Часов практической подготовки:

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)	17	10	7
Курсовое проектирование (КП, КР)	20		20
Подготовка к текущему контролю	15	8	7
Подготовка к промежуточной аттестации (ПА)	20	10	10
Всего:	72	41	31

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://nsu.ru/xmlui/bitstream/handle/nsu/9058/kr.pdf	Брайан У. Керниган, Деннис М. Ритчи. Язык программирования C	
http://cppstudio.com/wp-content/files/straustrup2.pdf	Бьерн Страуструп. Язык программирования C++.	

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://ravesli.com/uroki-cpp/	Ravesli. Уроки программирования на языке C++

8. Перечень информационных технологий

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

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
1	Интегрированная среда программирования Visual Studio Communication (распространяется без лицензии)
2	Компилятор DEV C++ (свободно распространяемый)

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

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

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

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Лекционная аудитория	
2	Компьютерный класс ВЛ ФПТИ	

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; Экзаменационные билеты; 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	Изложите основные принципы разработки программ.	ОПК-2.3.1 ПК-3.3.3
2	Опишите семь этапов программирования	ОПК-2.3.1 ПК-3.3.3
3	Как нужно использовать ГОСТ на оформление программной документации	ОПК-2.У.1

4	Приведите алгоритмы базовых управляющих структур	ОПК-7.У.1 ПК-3.3.3
5	Приведите алгоритмы дополнительных управляющих структур	ПК-3.3.3
6	Приведите пример решения задач с использованием алгоритма ветвящейся структуры	ПК-8.3.1
7	Объясните, какие существуют способы алгоритмизации циклических вычислительных процессов	ПК-3.У.3
8	Приведите пример решения задач с использованием алгоритма циклической структуры	ПК-3.3.3
9	Опишите структуру программ на языке С	ОПК-7.3.1
10	Перечислите основные элементы языка С	ПК-6.3.5
11	Перечислите типы данных в языке С/С++	ПК-6.3.6
12	Опишите базовые типы данных в языке С/С++.	ПК-6.3.6
13	Перечислите целочисленные типы	ПК-6.3.6
14	Объясните, в чем отличие целочисленных типов без знака	ПК-6.3.6
15	Дайте определение целочисленных литералов	ПК-6.3.6
16	Верно ли, что символьные типы относятся к целочисленным типам?	ПК-6.3.6
17	Существует ли в С тип bool ?	ПК-6.3.6
18	Расскажите, какие типы данных служат для представления вещественных чисел	ПК-6.3.6
19	Назовите основные правила работы с константами с плавающей точкой	ПК-6.3.6
20	Объясните, для чего необходимо преобразование типов	ПК-6.3.6
22	Объясните разницу между преобразованием типов и приведением типов	ПК-6.3.6
23	Перечислите составные типы данных в языке С/С++.	ПК-6.3.6
24	Приведите правила объявления и обработки массивов в С++11	ПК-6.В.1
25	Опишите основные особенности использования массивов в качестве аргументов функций	ПК-8.У.1
26	Объясните, чем отличается динамическое выделение памяти от статического	ПК-6.3.6
27	Продемонстрируйте на примере, чем отличается динамическое выделение памяти в С от С++	ПК-6.В.1
28	Расскажите про составной тип данных – строки в стиле С.	ПК-6.3.6
29	Продемонстрируйте на примере различие в использовании методов get() и getline()	ПК-6.В.1
30	Расскажите об основных функциях из библиотеки <string.h>	ПК-6.3.6
31	Объясните, какие проблемы смешивания цифрового и строкового ввода	ПК-8.У.1
32	Объясните, чем удобно использование класса string для работы со строками.	ПК-8.У.1
33	Расскажите о функциях и объектах класса string	ПК-8.У.1
34	Составные типы данных в языке С/С++. Структуры.	ПК-6.В.1

35	Расскажите про составной тип данных - структуры	ПК-8.У.1
36	Верно ли утверждение, что размер структуры можно вычислить, как сумму размеров ее полей	ПК-8.У.1
37	Опишите основные особенности использования структур в функциях	ПК-8.У.1
38	Приведите пример использования ссылочных переменных	ПК-6.В.1
39	Расскажите про составной тип данных –объединения	ПК-8.У.1
40	Расскажите про составной тип данных - перечисления	ПК-8.У.1
41	Перечислите операторы языка С	ПК-8.У.1
42	Приведите пример использования условного тернарного оператора	ПК-6.В.1
43	Расскажите об особенностях использования логических операторов в языке С	ПК-8.У.1
44	Перечислите побитовые операции, используемые в языке С	ОПК-7.В.1
45	Объясните разницу между методами класса iostream cin и cout и функциями printf() и scanf()	
46	Перечислите методы организации хранения переменных в памяти.	ОПК-7.В.1
47	Объясните различие автоматического и статического хранилищ.	ПК-8.У.1
48	Объясните связь схемы хранения и типа связывания.	ПК-8.У.1
49	При использовании перегрузки функций какой тип связывания (статическое или динамическое связывание) используется?	ПК-8.У.1
50	Правильно ли следующее утверждение: Для хранения статических переменных используется стек.	ПК-8.У.1
	Можно ли утверждать, что в однофайловых программах глобальные переменные идентичны статическим с внутренним связыванием	ОПК-7.В.1
51	Приведите особенности организации динамического хранилища	ПК-6.В.1
	3 семестр	
52	Объясните разницу между указателями и ссылками.	ПК-6.В.1
54	Объясните, в чем особенности использования const с указателями	ПК-6.В.1
55	Приведите пример использования функции с аргументами по умолчанию	ПК-6.В.1
56	Объясните, каким образом осуществляется перегрузка функций и зачем она нужна	ПК-8.У.1
57	Объясните, каким образом осуществляется перегрузка операторов и зачем она нужна	ПК-8.У.1
58	Перечислите основные признаки объектно-ориентированного программирования	ПК-8.У.1
59	Приведите пример создания класса	
60	Объясните смысл терминов «инкапсуляция», «полиморфизм»	ПК-6.В.1
61	Перечислите методы доступа к элементам класса	ПК-6.3.7
62	Объясните, чем отличается конструктор от других методов класса	ПК-6.3.7
63	Объясните разницу между объектом и классом	ПК-6.В.1
64	Объясните, чем отличаются функции-члены класса от данных-	ПК-6.В.1

	членов класса	
65	Перечислите, какие существуют конструкторы по умолчанию	ПК-6.В.1
66	Верно ли следующее утверждение «Наличие конструктора в классе не обязательно»	ОПК-7.В.1
67	Объясните смысл и назначение указателя this	ПК-6.В.1
68	Чему равен объем памяти, занимаемый объектом пустого класса?	ПК-6.В.1
69	Объясните, вызывается ли конструктор при передаче объекта в качестве параметра функции? А деструктор?	ПК-6.В.1
70	Объясните, зачем нужны приватные методы в классе?	ПК-3.3.3
71	Как вы думаете, можно ли сделать приватным конструктор	ПК-8.У.1
72	Объясните разницу между инициализацией копированием и uniform-инициализация переменных-членов в классе.	ПК-8.У.1
73	Как вы думаете, в каких случаях необходимо писать определение деструктора в классе.	ОПК-7.В.1
74	Объясните назначение дружественных функций	ПК-6.В.1
75	Приведите пример использования дружественных классов	ПК-6.В.1
76	Объясните понятие и назначение наследования	ПК-6.3.7
77	Перечислите виды наследования	ПК-8.У.1
78	Приведите пример полиморфного открытого наследования	ПК-8.У.1
79	Объясните, на что влияет уровень доступа при наследовании	ПК-8.У.1
80	Объясните отличие открытого наследования от закрытого и защищённого	ПК-8.У.1
81	Как вы думаете, могут ли наследоваться дружественные функции?	ПК-8.У.1
82	Как вы думаете, может ли имя поля производного класса совпадать с именем поля базового класса?	ПК-8.У.1
83	Объясните, что такое «виртуальные функции» и зачем они нужны	ПК-8.У.1
84	Как вы думаете, могут ли наследоваться виртуальные функции?	ПК-8.У.1
85	Объясните, для чего предназначены шаблоны? Какие виды шаблонов в C++ вы знаете	ПК-3.3.3
86	Объясните, в чем разница между определением и объявлением шаблона	ПК-8.У.1
87	Можно ли параметрам шаблона присваивать значения по умолчанию	ПК-8.У.1
88	Объясните, в чем состоят проблемы, возникающие при разделении шаблонного класса на интерфейс и реализацию?	ПК-3.3.3
89	Объясните, какие проблемы возникают при объявлении дружественной функции для класса-шаблона?	ПК-8.У.1
90	Поясните, чем отличается множественное наследование от простого	ПК-8.У.1
91	Как вы думаете, сколько может быть родительских классов при множественном наследовании	ПК-8.У.1
92	Как вы думаете, если производный класс не добавляет члены данных в базовый класс, то нужны ли конструкторы для производного класса?	ПК-3.3.3

93	Поясните, что такое контейнер и какие контейнеры используются в C++	ПК-8.У.1
94	Поясните, в чем отличие итератора от указателя	ПК-8.У.1
95	Нарисуйте структуру организации стека	ОПК-7.В.1
96	Объясните, в чем различие массивов и списков?	ПК-6.3.6
97	Перечислите, какие действия разрешены в стеке	ПК-6.3.6
98	Нарисуйте структуру организации очереди	ОПК-7.В.1
99	Поясните, сколько определений класса может быть в программе из нескольких файлов?	ПК-6.3.6
100	Объясните назначение «условной компиляции»	ОПК-2.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 выполнения курсовой работы
	Тема. Аппроксимация функций методом наименьших квадратов
1	Решение системы уравнений методом последовательных итераций
2	Решение системы уравнений методом Зейделя
3	Решение системы уравнений методом Гаусса
4	Решение системы уравнений методом обратной матрицы

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
	2 семестр	
1	<p>Что выведет программа на экран:</p> <pre>int i ; main() { while(i < 10) ;</pre>	ОПК-7.У.1 ОПК-7.В.1 ПК-8.3.1

	<pre>{ i++; } printf("%d", i); }</pre> <p>1) 10 2) неопределенное значение 3) 2 4) 1</p>	
2	<p>Формальным параметром функции может быть:</p> <p>1) указатель на указатель 2) указатель на функцию 3) структура 4) массив 5) функция</p>	ПК-6.3.5
3	<p>В языке С</p> <p>1) арифметические действия с данными различных числовых типов требуют явного преобразования типов 2) существуют специальные функции для получение кода символа 3) данные символьного типа char могут быть знаковыми и беззнаковыми 4) существует логический тип данных 5) существуют целочисленные данные знаковых и беззнаковых типов 6) все целочисленные переменные по умолчанию – беззнаковые</p> <p>Что будет напечатано?</p> <pre>#include void main() { short int a[] = {1,2,3}; printf("%d", sizeof(a)/sizeof(a[0])); }</pre> <p>1) 3 2) 4 3) 12 4) 6</p>	ПК-8.В.1
4	<p>Какой тип результата получится при сложении переменных типа short?</p> <p>1) double 2) int 3) short 4) long 5) float 6) unsigned</p>	ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.1
5	<p>Что будет выведено на экран кодом, показанным ниже?</p> <pre>int i = 4; int x = 6;</pre>	ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.1

	<pre>double z; z = x / i; printf("z=%.2f ", z); ?) z=1.00 ?) z=2.00 ?) z=NULL ?) z=0.00 ?) z=1.50</pre>	
6	<p>Что будет напечатано в результате выполнения программы?</p> <pre>#include void main() { int a = 5^3; float b = 1.5f; b += --a/2; printf("%.2f", b); }</pre> <p>1) 63.50 2) 64.00 3) 4.00 4) 3.50</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
7	<p>Пользователь вводит строку "Иванов Иван Иванович". Эту строку программа читает после нажатия клавиши Enter с помощью функции scanf("%s", str). Что будет занесено в массив str?</p> <p>1) "Иванов" 2) Ничего, так как неверно задан формат ввода строки 3) "Иванов Иван Иванович " 4) "Иванов Иван Иванович" 5) "Иванов " 6) Ничего, так как перед именем str не хватает символа &</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
8	<p>Что из перечисленного является управляющей последовательностью?</p> <p>1) \f 2) \” 3) \t 4) \ 5) \b</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
9	<p>Какой результат вернет функция strcmp("a","b"); ?</p> <p>1) Минус один 2) Плюс один 3) Отрицательное число 4) Положительное число</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
10	<p>Какие проблемы могут возникнуть при реализации фрагмента, показанного ниже, если функция malloc работает без ошибок?</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1</p>

	<pre>char *p = (char *)malloc(1000); ... realloc(p, 2000);</pre> <p>?) Произойдет потеря выделенного функцией malloc блока памяти ?) Фрагмент будет работать нормально ?) Произойдет потеря выделенного функцией realloc блока памяти ?) Произойдет выход за рамки выделенного блока памяти</p>	ОПК-7.У.
11	<p>Какие проблемы могут возникнуть при реализации фрагмента, показанного ниже, если функция malloc работает без ошибок?</p> <pre>char *p = (char *)malloc(1000); for(i = 0; i < 1000; i++) *p++ = 0; ... free(p);</pre> <p>1) Все элементы символьного массива будут содержать неопределенные значения, кроме первого 2) Произойдет потеря выделенного блока памяти 3) Фрагмент будет работать правильно 4) Произойдет выход за рамки выделенного блока памяти</p>	ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.
12	<p>Какую и сколько из строк case следует удалить, чтобы оставшаяся конструкция стала синтаксически корректной?</p> <pre>switch(a) { case 1: a++; break; case 2: case 3: a--; return; case 4: case 5: a+=1; continue; }</pre> <p>1) четвертую и пятую 2) вторую и четвертую 3) все и так нормально 4) вторую, четвертую и пятую 5) пятую</p>	ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.
13	<p>Какие операций из приведенных ниже корректны?</p> <p>1) 16<<<2 2) 16>>-2 3) 16>>2 4) 16<<+2 5) -16<<2 6) 16<<=2</p>	ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.
14	<p>После вызова функции выделения динамической памяти <code>int * p = malloc(1000 * sizeof(int));</code> какими операторами можно проверить факт успешности этой операции?</p> <p>1) if(p != 0)</p>	ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.

	<p>2) if(p != NULL) 3) if(p == NULL) 4) if(p == 0) 5) if(!p) 6) if(p)</p>	
15	<p>Что будет содержать массив x после выполнения кода, показанного ниже?</p> <pre>int x[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }; int y = 20; int i = 10; x[--i] = y;</pre> <p>1) { 20, 2, 3, 4, 5, 6, 7, 8, 9, 10 } 2) { 1, 2, 3, 4, 5, 6, 7, 8, 9, 20 } 3) { 1, 2, 3, 4, 5, 6, 7, 8, 20, 10 } 4) Произойдет выход за рамки массива 5) { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
16	<p>Возвращаемым значением функции может быть</p> <p>1) структура 2) указатель на указатель 3) указатель на функцию 4) функция 5) массив</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
17	<p>С помощью какого формата можно ввести вещественную переменную типа double функцией scanf()?</p> <p>1) %LF 2) %le 3) %d 4) %lf 5) %f 6) %df</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
18	<p>Чему равно значение переменной "x" после выполнения цикла for(i=0; i <10; x=i, i++) ?</p> <p>1) Зависит от начального значения переменной "x" 2) Равно одиннадцати 3) Равно десяти 4) Равно девяти</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
19	<p>С какими типами данных производятся побитовые операции:</p> <p>1) целочисленными 2) любыми числовыми 3) беззнаковыми числовыми 4) битовыми полями 5) любыми</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
20	<p>Как правильно обратиться к пятому элементу массива: double data[10]; ?</p> <p>1) *(&data[0]+4)</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1</p>

21	<p>2) 4[data] 3) *(&data+4) 4) data[5] 5) (&data+8)</p> <p>Что будет напечатано?</p> <pre>#include #include void main() { char a[] = "123"; printf("%d", strlen(a)); }</pre> <p>1) 8 2) 4 3) 3 4) 6</p>	<p>ОПК-7.У.</p> <p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
22	<p>Что будет напечатано в результате выполнения программы?</p> <pre>#include #define MAC1(a) (((a)<0)? (-a) : (a)) void main() { int V = 1; printf("%d", (int)MAC1(V-6.8)); }</pre> <p>1) 5 2) 6 3) -7 4) -8</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
23	<p>Что будет напечатано?</p> <pre>#include void main() { char a[] = {'1','2','3'}; printf("%d", sizeof(a)); }</pre> <p>?) 2 ?) 4 ?) 1 ?) 3</p>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
24	<p>Какую часть квадратной матрицы n*n заполнит приведенный фрагмент кода?</p> <pre>for(i=0;i<n;++i) for(j=0;j<i;++j) a[i][j]=i+j+1;</i;++j)</pre>	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>

	<p></n;++i)</p> <ol style="list-style-type: none"> 1) элементы под дополнительной диагональю 2) элементы над дополнительной диагональю 3) элементы на дополнительной диагонали 4) элементы под главной диагональю 5) элементы над главной диагональю 6) элементы на главной диагонали 	
25	<p>Может ли оператор continue завершать ветвь case оператора switch, предотвращая, подобно оператору break, выполнение следующей ветви case?</p> <ol style="list-style-type: none"> ?) только если оператор цикла вложен в оператор switch ?) нет, не может ?) только если оператор switch вложен в оператор цикла ?) да, может 	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
26	<p>Директивы препроцессора помещаются:</p> <ol style="list-style-type: none"> ?) вне определений функций ?) в начале файла ?) в начале строки, исключая предшествующие пробельные символы ?) в любом месте текста ?) в начале строки 	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
27	<p>Выберите верные утверждения относительно аргументов функций?</p> <ol style="list-style-type: none"> 1) Для того, чтобы изменить значение какой-либо переменной вызывающей функции внутри вызываемой функции, необходимо передавать адрес этой переменной в качестве фактического аргумента, а принимать этот адрес следует при помощи формального аргумента - указателя 2) При вызове функции нельзя изменить значение фактического аргумента, изменяя значение соответствующего ему формального аргумента 3) Можно определять значения аргументов по умолчанию 4) Если функция не имеет аргументов, то круглые скобки при вызове такой функции можно не писать 	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
28	<p>Какие операции допустимы с указателями?</p> <ol style="list-style-type: none"> 1) Сложение с целым числом 2) Присваивание 3) Сложение с другим указателем 4) Равно 5) Умножение 	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>
29	<p>Знак подчеркивания в начале имен функций _getch, _strdup и др. означает, что:</p> <ol style="list-style-type: none"> 1) Функции являются устаревшими и оставлены для совместимости с предыдущими версиями языка С 2) Функции реализованы как макроопределения через другие функции 3) Функции используют динамическое выделение памяти 	<p>ПК-6.3.5 ПК-8.В.1 ОПК-7.3.1 ОПК-7.У.</p>

30	<p>4) Функции отсутствуют в стандарте ANSI C</p> <p>Результат операции сравнения "a" > "b"?</p> <p>1) Операция не определена 2) Ложь 3) Зависит от типов a, b 4) Истина 5) Зависит от значений a, b</p>	ПК-6.3.5 ПК-8.B.1 ОПК-7.3.1 ОПК-7.У.
3 семестр		
1	<p>Верны ли утверждения?</p> <p>1) Объекты классов с конструкторами могут быть инициализированы при помощи списков инициализаторов, задаваемых в круглых скобках</p> <p>2) Объекты классов с конструкторами могут быть инициализированы, используя знак равенства, за которым следует отдельное значение</p>	ПК-6.3.6 ПК-6.B.1 ОПК-7.3.1 ОПК-7.У. ПК-8.B.1
2	<p>Подберите правильный ответ</p> <p>1) 1 - нет, 2 - да 2) 1 - нет, 2- нет 3) 1 - да, 2 - да 4) 1- да, 2 - нет</p>	ПК-6.3.6 ПК-6.B.1 ОПК-7.3.1 ОПК-7.У. ПК-8.B.1
3	<p>Верны ли утверждения?</p> <p>1) Агрегация – включение объектов одного класса в состав другого класса</p> <p>2) Ассоциация – двусторонняя связь классов, устанавливаемая между самостоятельными объектами</p> <p>Подберите правильный ответ</p> <p>1) 1 - да, 2 - нет 2) 1 - нет, 2 - да 3) 1 - да, 2 - да 4) 1 - нет, 2 - нет</p>	
4	<p>Верны ли утверждения?</p> <p>1) В языке C++ не нужно вызывать конструктор и деструктор предка, конструктор должен быть виртуальным</p> <p>2) В языке C++ не нужно вызывать конструктор и деструктор предка, деструктор должен быть виртуальным</p> <p>Подберите правильный ответ</p> <p>1) 1 - нет, 2 - нет 2) 1 - да, 2 - да 3) 1 - нет, 2 - да 4) 1 - да, 2 - нет</p>	
5	<p>Верны ли утверждения?</p> <p>1) Деструктор – метод, вызываемый автоматически во время создания объекта соответствующего класса с целью его</p>	ПК-6.3.6 ПК-6.B.1 ОПК-7.3.1 ОПК-7.У. ПК-8.B.1

6	<p>инициализации</p> <p>2) Конструктор - метод, вызываемый автоматически во время создания объекта соответствующего класса с целью его инициализации</p> <p>Подберите правильный ответ</p> <p>1) 1 - да, 2 - да 2) 1 - да, 2 - нет 3) 1 - нет, 2 - да 4) 1 - нет, 2 - нет</p> <p>Верны ли утверждения?</p> <p>1) Если оператор New не может выделить запрашиваемый объем памяти, он возвратит NULL-указатель, который содержит значение 0</p> <p>2) Если оператор New не может выделить запрашиваемый объем памяти, он возвратит NULL-указатель, который содержит значение 1</p> <p>Подберите правильный ответ</p> <p>1) 1 - да, 2 - да 2) 1- да, 2 - нет 3) 1 - нет, 2 - нет 4) 1 - нет, 2 – да</p>	
7	<p>Верны ли утверждения?</p> <p>1) Имя деструктора должно начинаться со знака тильды (~), за которым следует имя класса</p> <p>2) Деструктор не должен иметь параметров</p> <p>Подберите правильный ответ</p> <p>1) 1- нет, 2 - да 2) 1 - да, 2 - да 3) 1 - да, 2 - нет 4) 1- нет, 2 - нет</p>	<p>ПК-6.3.6 ПК-6.В.1 ОПК-7.3.1 ОПК-7.У. ПК-8.В.1</p>
8	<p>Верны ли утверждения?</p> <p>1) Инкапсуляция - свойство языка программирования, позволяющее объединить и защитить данные и код в объект и скрыть реализацию объекта от пользователя</p> <p>2) Наследование – отношение между классами, обеспечивающее возможность конструирования новых, более сложных классов из уже имеющихся посредством добавления полей и определения новых методов</p> <p>Подберите правильный ответ</p> <p>1) 1 - да, 2 - нет 2) 1 - нет, 2 - нет 3) 1 - да, 2 - да 4) 1- нет, 2 – да</p>	
9	<p>Верны ли утверждения?</p> <p>1) Перегрузка используется в ООП при необходимости задания различных реализаций одноименных методов в иерархии классов</p> <p>2) Переопределение используется в ООП при необходимости</p>	

	<p>задания различных реализаций одноименных методов в иерархии классов Подберите правильный ответ 1) 1 - да, 2 - нет 2) 1 - нет, 2 - нет 3) 1 - нет, 2 - да 4) 1 - да, 2 - да</p> <p>10 Верны ли утверждения? 1) Перегрузка используется в ООП, когда определяется несколько функций с одним именем 2) Переопределение используется в ООП, когда определяется несколько функций с одним именем Подберите правильный ответ 1) 1 - да, 2 - да 2) 1 - нет, 2 - нет 3) 1 - нет, 2 - да 4) 1 - да, 2 - нет</p> <p>11 Изменить реализацию объекта без модификации программы, если его интерфейс остался прежним, позволяет 1) наследование 2) инкапсуляция 3) конструктор 4) полиморфизм</p> <p>12 Свойство ООП, позволяющее использовать один и тот же интерфейс для различных действий, называется 1) конструктором 2) инкапсуляцией 3) полиморфизмом 4) наследованием</p> <p>13 Создавать иерархии объектов позволяет 1) инкапсуляция 2) конструктор 3) наследование 4) полиморфизм</p> <p>14 Упорядоченный набор элементов данных, в котором можно удалять и добавлять элементы, причем новый элемент всегда записывается в его конец, а очередной читаемый или удаляемый элемент также выбирается из его конца, называется 1) очередью 2) списком 3) массивом 4) стеком</p> <p>15 Для обращения к объектно-ориентированному вводу/выводу потоком программа на Си++ должна включать 1) файл ioscanf 2) класс iostream</p>	<p>ПК-6.3.6 ПК-6.В.1 ОПК-7.3.1 ОПК-7.У. ПК-8.В.1</p>
--	--	--

	3) класс ioscanf 4) файл iostream.h.	
16	Из перечисленного: 1) cin; 2) cout; 3) cerr; 4) clog - с стандартными потоками Си++ относятся 1) 3, 4 2) 3, 4, 5 3) 1, 2 4) 1, 2,5	
17	Из перечисленного: 1) ostream; 2) stringstream; 3) istream - к классам, порожденным от класса ios, можно отнести к классам 1) 2 и 3 2) 1 и 3 3) только 1 4) только 2	
18	Из перечисленного: 1) прототипы функций; 2) переменные в описании класса; 3) функции-компоненты - к понятию «метод в Си++» можно отнести 1) 1 и 3 2) 2 и 3 3) только 1 4) только 3	ПК-6.3.6 ПК-6.В.1 ОПК-7.3.1 ОПК-7.У. ПК-8.В.1
19	Класс в Си++ - это 1) способ описания однотипной группы программных модулей 2) новый тип данных 3) способ описания отдельного программного модуля 4) объект	
20	Класс с как минимум одной чистой виртуальной функцией называется 1) производным 2) дружественным 3) абстрактным 4) базовым	
21	Ключевое слово friend (дружественный) используется для того, чтобы 1) обеспечить доступ из абстрактных классов 2) обеспечить доступ порожденных классов к компонентам базовых 3) обеспечить доступ к данным, объявленным, как private 4) обеспечить доступ к данным, объявленным как portable	
22	Компоненты называются статическими 1) при размещении в разделе private 2) при наличии соответствующего спецификатора класса памяти 3) при размещении в разделе protected 4) по умолчанию	
23	Конструктором по умолчанию называется такой конструктор,	

	<p>который</p> <ol style="list-style-type: none"> 1) принимает бесконечное множество аргументов 2) либо принимает один аргумент, либо не принимает ни одного аргумента 3) принимает не более одного аргумента 4) не принимает никаких аргументов 	
24	<p>Конструкторы отличаются от прочих функций-компонентов тем, что</p> <ol style="list-style-type: none"> 1) могут иметь аргументы по умолчанию 2) имеют возможность иметь атрибут virtual 3) могут не принимать аргументы 4) имеют то же самое имя, что и класс, к которому они относятся 	
25	<p>Объект в Си++ (в смысле ООП) - это</p> <ol style="list-style-type: none"> 1) ряд выражений, которые определяют реакцию на управляющее воздействие 2) специальный символ, идентификатор или ключевое слово с параметрами 3) реализация объектно-ориентированного модуля 4) переменная типа className, где className - определенный ранее класс 	
26	<p>Операция new в Си++ выполняет</p> <ol style="list-style-type: none"> 1) динамическое распределение памяти 2) динамического резервирования объектов 3) создание нового объекта 4) создание нового функционального модуля 	<p>ПК-6.3.6 ПК-6.В.1 ОПК-7.3.1 ОПК-7.У. ПК-8.В.1</p>
27	<p>Перегрузкой называется определение нескольких функций, имеющих</p> <ol style="list-style-type: none"> 1) одинаковое имя, но разные списки аргументов 2) разные списки аргументов 3) одинаковое имя, но разный контекст 4) одинаковое имя в иерархии классов 	
28	<p>Переопределение функций и операций называется</p> <ol style="list-style-type: none"> 1) виртуализацией 2) перегрузкой 3) сменой контекста 4) наследованием 	
29	<p>Свойство позволяющее использовать один и тот же интерфейс для общего класса действия, называется</p> <ol style="list-style-type: none"> 1) наследование 2) абстракция 3) полиморфизм 4) инкапсуляция 	

30	Функция не может возвращать 1) комплексную переменную 2) указатель на массив 3) массив 4) структуру	
----	--	--

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:

- Изложение материала по рассматриваемой теме;

- Демонстрация примеров решения конкретных задач;
- Ответы на возникающие вопросы по теме лекции.

11.2. Методические указания для обучающихся по участию в семинарах

Учебным планом не предусмотрено

11.3. 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

Требования к проведению практических занятий, темы занятий и задания приведены в методических материалах, расположенных на сервере локальной сети кафедры 2.

11.4. 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.

Методические указания, задания, структура и форма отчета, а также примеры выполнения лабораторных работ изложены в методических указаниях:

1. С.Л.Козенко, В.А.Галанина. Информатика.Лабораторный практикум. ч.3/ Учебно-методическое пособие – СПб, ГУАП, 2020. – 67 с.

2. С.Л.Козенко, В.А.Галанина. Информатика.Лабораторный практикум. ч.4/ Учебно-методическое пособие – СПб, ГУАП, 2021. –60с.

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

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

Курсовой проект/ работа позволяет обучающемуся:

- систематизировать и закрепить полученные теоретические знания и практические умения
- применить полученные знания, умения и практический опыт при решении практической задачи обработки результатов эксперимента
- углубить теоретические знания в соответствии с заданной темой;
- развить системное мышление, творческую инициативу, самостоятельность, организованность и ответственность за принимаемые решения;
- сформировать навыки планомерной регулярной работы над решением поставленных задач.

Методические указания, задания, структура и форма пояснительной записки, а также примеры выполнения курсовой работы изложены в методических указаниях:

Галанина В.А., Козенко С.Л. Информатика. Методические указания к выполнению курсовой работы, ч.1.- СПб, ГУАП, 2018. – 55 с;

Галанина В.А., Козенко С.Л. Информатика. Методические указания к выполнению курсовой работы, ч.2.- СПб, ГУАП, 2019. – 47 с;

11.6. 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 обучающихся являются:

- учебно-методический материал по дисциплине;

11.7. 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.

Текущий контроль осуществляется трижды в семестре по результатам выполнения промежуточных этапов выполнения курсовой работы и оценивается в баллах. Эти баллы добавляются к баллам, полученным при защите курсовой работы и учитываются при выставлении окончательной оценки.

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

Для допуска к экзамену студентам необходимо выполнить все лабораторные работы, предусмотренные планом. Работы выполняются и защищаются преподавателю в соответствии с графиком сдачи работ. В случае сдачи работы после указанного срока, снижается максимально возможное количество баллов за данную работу.

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