МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Директор департамента образования
_____ П. Е. Троян
«___» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Информационные технологии

Уровень образования: **высшее образование - бакалавриат**
Направление подготовки / специальность: **11.03.04 Электроника и наноэлектроника**
Направленность (профиль) / специализация: **Промышленная электроника**
Форма обучения: **заочная**
Факультет: **ЗиВФ, Заочный и вечерний факультет**
Кафедра: **ПрЭ, Кафедра промышленной электроники**
Курс: **1, 2**
Семестр: **1, 2, 3**
Учебный план набора 2015 года

Распределение рабочего времени

№	Виды учебной деятельности	1 семестр	2 семестр	3 семестр	Всего	Единицы
1	Лекции	10	0	0	10	часов
2	Практические занятия	6	0	2	8	часов
3	Лабораторные работы	0	8	4	12	часов
4	Всего аудиторных занятий	16	8	6	30	часов
5	Самостоятельная работа	56	96	93	245	часов
6	Всего (без экзамена)	72	104	99	275	часов
7	Подготовка и сдача экзамена / зачета	0	4	9	13	часов
8	Общая трудоемкость	72	108	108	288	часов
					8.0	З.Е.

Контрольные работы: 2 семестр - 2; 3 семестр - 2
Зачет: 2 семестр
Экзамен: 3 семестр

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шелупанов А.А.
Должность: Ректор
Дата подписания: 23.08.2017
Уникальный программный ключ:
c53e145e-8b20-45aa-9347-a5e4dbb90e8d

Томск 2018

Согласована на портале № 36862

Рассмотрена и одобрена на заседании кафедры
протокол № 50 от «26» 4 2018 г.

ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа дисциплины составлена с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 11.03.04 Электроника и наноэлектроника, утвержденного 12.03.2015 года, рассмотрена и одобрена на заседании кафедры ПрЭ «___» _____ 20__ года, протокол №_____.

Разработчики:

зав. кафедрой, профессор каф. ПрЭ	_____	С. Г. Михальченко
старший преподаватель каф. ПрЭ	_____	Д. А. Савин

Заведующий обеспечивающей каф. ПрЭ	_____	С. Г. Михальченко

Рабочая программа дисциплины согласована с факультетом и выпускающей кафедрой:

Декан ЗиВФ	_____	И. В. Осипов
Заведующий выпускающей каф. ПрЭ	_____	С. Г. Михальченко

Эксперты:

Зам. зав. кафедрой ПРЭ по методической работе, профессор каф. ПРЭ	_____	Н. С. Легостаев
Доцент кафедры физической электроники (ФЭ)	_____	И. А. Чистоедова

Согласована на портале № 36862

1. Цели и задачи дисциплины

1.1. Цели дисциплины

Целью преподавания дисциплины «Информационные технологии» является изучение основ современных способов обработки информации с использованием средств вычислительной техники, знакомство с популярными программными продуктами, применяемыми как в инженерных расчетах, так в офисных технологиях.

1.2. Задачи дисциплины

– Задачи изучения дисциплины «Информационные технологии» состоят в последовательном освоении материала по основам вычислительной техники и по ее применению в решении учебных и исследовательских задач. В ходе изучения дисциплины «Информационные технологии» студенты знакомятся с устройством, принципами работы компьютера, получают представление о системном и прикладном программном обеспечении, овладевают навыками практической работы с компонентами MS Office. К задачам освоения дисциплины относится изучение пакета математических расчетов MathCAD и основ программирования на языке C++.

2. Место дисциплины в структуре ОПОП

Дисциплина «Информационные технологии» (Б1.Б.8) относится к блоку 1 (базовая часть).

Предшествующими дисциплинами, формирующими начальные знания, являются: Информационные технологии, Аппаратное и программное обеспечение ЭВМ.

Последующими дисциплинами являются: Информационные технологии, Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты, Математическое моделирование и программирование, Профессиональные математические пакеты.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

– ОПК-6 способность осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий;

– ОПК-7 способностью учитывать современные тенденции развития электроники, измерительной и вычислительной техники, информационных технологий в своей профессиональной деятельности;

– ОПК-9 способностью использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности;

В результате изучения дисциплины обучающийся должен:

– **знать** принципы работы персональных компьютеров, идеологию современных операционных систем, основные методы разработки алгоритмов и программ, процесс компиляции и создания программного продукта, способы размещения в памяти компьютера структур данных, используемых для представления типовых информационных объектов и обращения к ним, типовые алгоритмы обработки данных, принципы использования прикладных математических пакетов.

– **уметь** давать стратегическую оценку решаемой технической задачи, основывающуюся на понимании и ясном представлении целей исследований и абстрагировании от шаблонов и алгоритмов поиска решений; решать задачи обработки данных с помощью современных инструментальных средств конечного пользователя; создавать, отлаживать и тестировать программы на платформеннонезависимом языке высокого уровня (C++); представлять результаты исследований в удобном формате (MsOffice).

– **владеть** инструментами создания современных программных продуктов - отладочными средами высокого уровня; навыками написания, тестирования и отладки программных средств; навыками постановки научно-технических задач, выбора методов их решения, получения и предоставления результатов, интерпретации и анализа полученных данных.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 8.0 зачетных единицы и представлена в таблице 4.1.

Согласована на портале № 36862

Таблица 4.1 – Трудоемкость дисциплины

Виды учебной деятельности	Всего часов	Семестры		
		1 семестр	2 семестр	3 семестр
Аудиторные занятия (всего)	30	16	8	6
Лекции	10	10	0	0
Практические занятия	8	6	0	2
Лабораторные работы	12	0	8	4
Самостоятельная работа (всего)	245	56	96	93
Оформление отчетов по лабораторным работам	104	0	64	40
Проработка лекционного материала	28	28	0	0
Подготовка к практическим занятиям, семинарам	47	28	0	19
Выполнение контрольных работ	66	0	32	34
Всего (без экзамена)	275	72	104	99
Подготовка и сдача экзамена / зачета	13	0	4	9
Общая трудоемкость, ч	288	72	108	108
Зачетные Единицы	8.0			

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Разделы дисциплины и виды занятий приведены в таблице 5.1.

Таблица 5.1 – Разделы дисциплины и виды занятий

Названия разделов дисциплины	Лек., ч	Прак. зан., ч	Лаб. раб., ч	Сам. раб., ч	Всего часов (без экзамена)	Формируемые компетенции
1 семестр						
1 Средства обработки информационных ресурсов. Понятие алгоритма. Обращение к данным. Алгоритмические конструкции. Блок-схемы. Операторы. Обзор вычислительных математических пакетов. Принципы обращения к ячейкам памяти. Адрес переменной. Указатели.	4	4	0	28	36	ОПК-6, ОПК-7, ОПК-9
2 Подпрограммы: заголовок, описание, реализация алгоритма, вызов подпрограммы, передача параметров в тело функции. Возможности представления информации. Типы файлового хранения, расширения файлов. Стандартные подпрограммы обработки данных. MsOffice.	6	2	0	28	36	ОПК-6, ОПК-7, ОПК-9
Итого за семестр	10	6	0	56	72	

Согласована на портале № 36862

2 семестр						
3 Массивы. Статические и динамические объекты, адресация. Одномерные и двумерные массивы. Работа с символьными массивами. Потоковый ввод-вывод. Динамические строки.	0	0	4	48	52	ОПК-6, ОПК-7, ОПК-9
4 Файловая система. Программная реализация. Дескриптор файла. Доступ к данным. Информационная безопасность.	0	0	4	48	52	ОПК-6, ОПК-7, ОПК-9
Итого за семестр	0	0	8	96	104	
3 семестр						
5 Структурированные типы данных, битовые поля, объединения. Приемы программирования микроконтроллеров, битовые операции.	0	2	2	51	55	ОПК-6, ОПК-7, ОПК-9
6 Понятие класса. Поля и методы класса. Спецификаторы доступа. Статические и динамические объекты класса.	0	0	2	42	44	ОПК-6, ОПК-7, ОПК-9
Итого за семестр	0	2	4	93	99	
Итого	10	8	12	245	275	

5.2. Содержание разделов дисциплины (по лекциям)

Содержание разделов дисциплин (по лекциям) приведено в таблице 5.2.

Таблица 5.2 – Содержание разделов дисциплин (по лекциям)

Названия разделов	Содержание разделов дисциплины (по лекциям)	Трудоемкость, ч	Формируемые компетенции
1 семестр			
1 Средства обработки информационных ресурсов. Понятие алгоритма. Обращение к данным. Алгоритмические конструкции. Блок-схемы. Операторы. Обзор вычислительных математических пакетов. Принципы обращения к ячейкам памяти. Адрес переменной. Указатели.	Понятие информации. Предмет информатики. Информационные процессы. Информатизация общества. Компьютеры. Технология проектирования и отладки программ. Инструментальные средства программирования. Основные принципы создания программного обеспечения. Основы программирования на языке верхнего уровня (на примере С++). Процесс создания программного кода. Отладка программ. Трассировка программного кода.Алгоритмические конструкции. Операторы выбора. Перечисляемый тип данных. Операторы цикла. Использование переменных логического типа. Оператор безусловного перехода. Организация диалога с пользователем.	2	ОПК-6, ОПК-7, ОПК-9
	Принципы обращения к ячейкам памяти. Адрес переменной. Указатели.Статическое и динамическое распределение памяти. Функции динамического распределения памяти. Освобождение памя-	2	

Согласована на портале № 36862

	ти. Операторы new и delete. Обращение к переменным "по имени" и "по адресу". Отличие указателя от динамической переменной.Подпрограммы: процедуры и функции. Структура подпрограммы. Формальные и фактические параметры. Передача параметров в тело функции. Вызов подпрограммы. Возвращение значения, оператор return. Область видимости переменной.		
	Итого	4	
2 Подпрограммы: заголовок, описание, реализация алгоритма, вызов подпрограммы, передача параметров в тело функции. Возможности представления информации. Типы файлового хранения, расширения файлов. Стандартные подпрограммы обработки данных. MsOffice.	Описание и вызов подпрограммы. Прототип подпрограммы. Адрес функции, указатель на подпрограмму. Область видимости переменной. Глобальные и локальные переменные. Перегрузка функций.Массивы - хранилище однотипных данных. Одномерные (вектора) и многомерные (матрицы) массивы. Описание массива, обращение к элементам массива по индексу. Генератор случайных чисел. Работа с массивами в цикле.Статические и динамические массивы. Функции calloc(), malloc() и free(). Адрес массива = адрес начальной ячейки. Указатель на массив. Передача массива в функцию. Динамические одномерные и двумерные массивы. Обращение к элементам массива "адрес+смещение".	2	ОПК-6, ОПК-7, ОПК-9
	Двумерные статические и динамические. Три способа размещения в памяти двумерных массивов. Освобождение памяти. Обращение к ячейкам двумерного массива.Работа с символьными массивами. Потоковый ввод-вывод. Динамические строки. Строки символов. Строка – массив символов. Вывод строки на экран. Последний элемент строки. Указатель на строку.Копирование и сравнение строк. Статические и динамические строки. Функции преобразования строковых типов данных.- Файловые подсистемы ОС, хранение данных на диске. Дескриптор файла в программе. Доступ к файлу, совместный доступ, транзакции. Файловые операции (связать, открыть, закрыть, читать, писать, определить конец).Работа с файлами при помощи потоков ввода-вывода. Библиотека . Файловые операции. Поиск в файле.	2	
	Структуры. Описание в программе, доступ к полям структуры. Размещение в памяти. Указатели на структуру. Статические и динамические переменные структурного типа. Битовые поля. Объединения. Статические и динамические переменные структурного типа.Приемы программирования микроконтроллеров, битовые операции. Поразрядные логические операции. Поразрядные операции сдвига. Обращение к разрядам при помощи битовых полей.Введение в понятие "класс". Поля и методы класса. Спецификаторы доступа, использова-	2	

Согласована на портале № 36862

	ние set- и get- методов. Конструктор и деструктор.Указатель на объект класса. Статические и динамические объекты класса. Конструктор и деструктор для статических и динамических объектов класса. Отделение интерфейса программы от реализации. Дружественные функции. Перегрузка операторов. Наследование.Вопросы информационной безопасности. Угрозы информационных атак. Компьютерные вирусы.		
	Итого	6	
Итого за семестр		10	
Итого		10	

5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами

Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами представлены в таблице 5.3.

Таблица 5.3 – Разделы дисциплины и междисциплинарные связи

Наименование дисциплин	№ разделов данной дисциплины, для которых необходимо изучение обеспечивающих и обеспечиваемых дисциплин					
	1	2	3	4	5	6
Предшествующие дисциплины						
1 Информационные технологии	+	+	+	+	+	+
2 Аппаратное и программное обеспечение ЭВМ	+	+	+	+	+	+
Последующие дисциплины						
1 Информационные технологии			+	+	+	+
2 Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты	+	+	+	+	+	
3 Математическое моделирование и программирование	+	+	+	+	+	+
4 Профессиональные математические пакеты	+	+	+	+	+	

5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий представлено в таблице 5.4.

Таблица 5.4 – Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Компетенции	Виды занятий				Формы контроля
	Лек.	Прак. зан.	Лаб. раб.	Сам. раб.	

Согласована на портале № 36862

ОПК-6	+	+	+	+	Контрольная работа, Отчет по индивидуальному заданию, Проверка контрольных работ, Отчет по лабораторной работе, Опрос на занятиях, Тест
ОПК-7	+	+	+	+	Контрольная работа, Отчет по индивидуальному заданию, Проверка контрольных работ, Отчет по лабораторной работе, Опрос на занятиях, Тест
ОПК-9	+	+	+	+	Контрольная работа, Отчет по индивидуальному заданию, Проверка контрольных работ, Отчет по лабораторной работе, Опрос на занятиях, Тест

6. Интерактивные методы и формы организации обучения

Не предусмотрено РУП.

7. Лабораторные работы

Наименование лабораторных работ приведено в таблице 7.1.

Таблица 7.1 – Наименование лабораторных работ

Названия разделов	Наименование лабораторных работ	Трудоемкость, ч	Формируемые компетенции
2 семестр			
3 Массивы. Статические и динамические объекты, адресация. Одномерные и двумерные массивы. Работа с символьными массивами. Потоковый ввод-вывод. Динамические строки.	Научиться создавать подпрограммы и вызывать их из тела основной функции. Освоить применение оператора return, возвращающего значение функции. Научиться передавать параметры в функцию и корректно возвращать значения – результаты вычислений из функции. Освоить передачу параметров в функцию по значению, по ссылке и по указателю. Научиться создавать прототипы функций, освоить перегрузку функций. Изучить функции библиотеки math.h. Научиться пользоваться средствами пошаговой трассировки кода и просмотра текущих значений переменных.Контрольная работа №1. Программа. Операторы. Ввод-вывод, Переменные. Адреса.	2	ОПК-6, ОПК-7, ОПК-9
	Научиться создавать массивы, выделять память под элементы массива и инициировать их значениями. Научиться обращаться к элементам массивов как при помощи оператора имя[индекс], так и посредством методики *(адрес+смещение). Освоить понятия: адрес массива, адрес элемента массива, смещение, индекс. Освоить пошаговую трассиров-	2	

Согласована на портале № 36862

	ку программы с массивами, научиться отражать в окне watch элементы массива и их адреса. Научиться передавать массив в функцию.		
	Итого	4	
4 Файловая система. Программная реализация. Дескриптор файла. Доступ к данным. Информационная безопасность.	Освоить способ динамического захвата и освобождения памяти под двумерные массивы данных (матрицы) – при помощи оператора new и посредством функций malloc( ) и calloc( ). Повторить понятия, операции и закрепить умения и навыки матричной алгебры. Уяснить практическую разницу и сходство в языке С++ понятий «массив» и «указатель». Повысить навыки отладки программного кода на С++, трассировки программы и просмотра значений переменных в окне Watch.Контрольная работа №2. Массив. Обращение к элементам массива. Указатель на массив. Адрес + смещение. Динамические массивы.	2	ОПК-6, ОПК-7, ОПК-9
	Научиться работать со строками. Освоить способ динамического захвата и освобождения памяти под символьные массивы. Уяснить разницу между понятиями «размер массива символов» и «длина строки» при помощи функций sizeof и strlen. Освоить функции библиотеки «string.h». Научится переводить числовые данные в строковые и обратно.Операционная система. Файловые таблицы. Обращение к оборудованию ПК. BIOS. Setup BIOS. Драйверы.	2	
	Итого	4	
Итого за семестр		8	
3 семестр			
5 Структурированные типы данных, битовые поля, объединения. Приемы программирования микроконтроллеров, битовые операции.	Научиться работать с файловыми переменными посредстваом подпрограмм библиотек fstream.h и stdio.h. Закрепить навыки работы со строками. Научиться открывать файлы для записи и чтения, создавать и уничтожать файлы, определять конец файла.Операционная система. Файловые таблицы. Элементы информационной безопасности.	2	ОПК-6, ОПК-7, ОПК-9
	Итого	2	
6 Понятие класса. Поля и методы класса. Спецификаторы доступа. Статические и динамические объекты класса.	Освоить умение работать с разрядами в байте. Научиться изменять произвольный бит в байте, не меняя остальных. Выводить на экран двоичный код переменной произвольного типа. Повысить понимание преобразований двоичного формата. Применять операции поразрядного сдвига и поразрядные логические операции, а также при помощи структур объединение (union) и битовые поля.Контрольная работа №4. Структурированные данные. Указатель на структуру. Динамические структуры. Объединение, битовые поля.	2	ОПК-6, ОПК-7, ОПК-9
	Итого	2	

Согласована на портале № 36862

Итого за семестр		4	
Итого		12	

8. Практические занятия (семинары)

Наименование практических занятий (семинаров) приведено в таблице 8.1.

Таблица 8.1 – Наименование практических занятий (семинаров)

Названия разделов	Наименование практических занятий (семинаров)	Трудоемкость, ч	Формируемые компетенции
1 семестр			
1 Средства обработки информационных ресурсов. Понятие алгоритма. Обращение к данным. Алгоритмические конструкции. Блок-схемы. Операторы. Обзор вычислительных математических пакетов. Принципы обращения к ячейкам памяти. Адрес переменной. Указатели.	Создание проекта на Visual C++, написать простейшую программу на языке C++, отладить и запустить на выполнение программу. Научиться задавать переменные и константы следующих типов: целого, вещественного, символьного и строкового. Освоить два способа ввода с клавиатуры и вывода на экран значений этих переменных – форматный и потоковый способы. Изучить спецификаторы, форматы ввода-вывода и управляющие символы.	2	ОПК-6, ОПК-7, ОПК-9
	Изучить операторы языка C++, научиться применять их при написании программ. Научиться отслеживать значения числовых и логических переменных и выражений в процессе отладки программы. Освоить умение переводить мнемонические выражения в последовательность операторов.	2	
	Итого	4	
2 Подпрограммы: заголовок, описание, реализация алгоритма, вызов подпрограммы, передача параметров в тело функции. Возможности представления информации. Типы файлового хранения, расширения файлов. Стандартные подпрограммы обработки данных. MsOffice.	Принципы обращения к ячейкам памяти. Адрес переменной. Указатели.Научиться работать со статическими и динамическими переменными. Освоить применение операций адресации и разадресации. Научиться выделять память под динамические переменные и освобождать ее при помощи операторов new и delete. Освоить обращение к динамическим объектам. Научиться выделять память под динамические переменные и освобождать ее при помощи функций calloc, malloc и free. Научиться визуализировать адреса статических и динамических объектов.	2	ОПК-6, ОПК-7, ОПК-9
	Итого	2	
Итого за семестр		6	
3 семестр			
5 Структурированные типы данных, битовые поля, объединения. Приемы	Освоить умение работать со структурами данных. Научиться создавать новый комбинированный тип данных – структуру. Научиться обращаться к статическим и динамическим переменным этого типа	2	ОПК-6, ОПК-7, ОПК-9

Согласована на портале № 36862

программирования микроконтроллеров, битовые операции.	и к массиву структур. Закрепить навыки работы с символьными строками.Контрольная работа №3. Строковые типы данных. Строка как массив. Строка как указатель. Динамические строки.		
	Итого	2	
Итого за семестр		2	
Итого		8	

9. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 9.1.

Таблица 9.1 – Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов	Виды самостоятельной работы	Трудоемкость, ч	Формируемые компетенции	Формы контроля
1 семестр				
1 Средства обработки информационных ресурсов. Понятие алгоритма. Обращение к данным. Алгоритмические конструкции. Блок-схемы. Операторы. Обзор вычислительных математических пакетов. Принципы обращения к ячейкам памяти. Адрес переменной. Указатели.	Подготовка к практическим занятиям, семинарам	14	ОПК-6, ОПК-7, ОПК-9	Опрос на занятиях, Отчет по индивидуальному заданию, Тест
	Проработка лекционного материала	14		
	Итого	28		
2 Подпрограммы: заголовок, описание, реализация алгоритма, вызов подпрограммы, передача параметров в тело функции. Возможности представления информации. Типы файлового хранения, расширения файлов. Стандартные подпрограммы обработки данных. MsOffice.	Подготовка к практическим занятиям, семинарам	14	ОПК-6, ОПК-7, ОПК-9	Опрос на занятиях, Отчет по индивидуальному заданию, Тест
	Проработка лекционного материала	12		
	Проработка лекционного материала	2		
	Итого	28		
Итого за семестр		56		
2 семестр				
3 Массивы. Статические и динамические	Выполнение контрольных работ	16	ОПК-6, ОПК-7,	Контрольная работа, Отчет по лабораторной ра-

Согласована на портале № 36862

объекты, адресация. Одномерные и двумерные массивы. Работа с символьными массивами. Потоковый ввод-вывод. Динамические строки.	Оформление отчетов по лабораторным работам	32	ОПК-9	боте, Проверка контрольных работ, Тест
	Итого	48		
4 Файловая система. Программная реализация. Дескриптор файла. Доступ к данным. Информационная безопасность.	Выполнение контрольных работ	16	ОПК-6, ОПК-7, ОПК-9	Контрольная работа, Отчет по лабораторной работе, Проверка контрольных работ, Тест
	Оформление отчетов по лабораторным работам	32		
	Итого	48		
Итого за семестр		96		
	Подготовка и сдача зачета	4		Зачет
3 семестр				
5 Структурированные типы данных, битовые поля, объединения. Приемы программирования микроконтроллеров, битовые операции.	Выполнение контрольных работ	16	ОПК-6, ОПК-7, ОПК-9	Контрольная работа, Отчет по индивидуальному заданию, Отчет по лабораторной работе, Проверка контрольных работ, Тест
	Подготовка к практическим занятиям, семинарам	19		
	Оформление отчетов по лабораторным работам	16		
	Итого	51		
6 Понятие класса. Поля и методы класса. Спецификаторы доступа. Статические и динамические объекты класса.	Выполнение контрольных работ	18	ОПК-6, ОПК-7, ОПК-9	Контрольная работа, Отчет по лабораторной работе, Проверка контрольных работ, Тест
	Оформление отчетов по лабораторным работам	24		
	Итого	42		
Итого за семестр		93		
	Подготовка и сдача экзамена	9		Экзамен
Итого		258		

10. Курсовой проект / курсовая работа

Не предусмотрено РУП.

11. Рейтинговая система для оценки успеваемости обучающихся

Рейтинговая система не используется.

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Михальченко Сергей Геннадьевич Информационное обеспечение задач расчета и проектирования электрических схем: моногр. / С. Г. Михальченко. – Томск: Изд-во Томск. гос. ун-та систем упр. и радиоэлектроники, 2016. – 200 с. : ил., табл. – Библиогр.: с. 192. [Электронный ресурс] - Режим доступа: http://ie.tusur.ru/docs/msg/io.rar (дата обращения: 04.07.2018).

12.2. Дополнительная литература

1. Информатика. Базовый курс: Учебник для вузов / С. В. Симонович [и др.] ; ред. С. В. Симонович. – 2-е изд. – СПб. : Питер, 2007. – 639 с. (наличие в библиотеке ТУСУР - 57 экз.)

Согласована на портале № 36862

2. Компьютерные технологии в науке, образовании и производстве электронной техники / Кручинин В.В., Тановицкий Ю.Н., Хомич С.Л. – Томск, 2012. – 154 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/training/publications/967 (дата обращения: 04.07.2018).

3. Саликаев, Юрий Рафаэльевич. Компьютерное моделирование и проектирование [Электронный ресурс] : учебное пособие / Ю. Р. Саликаев; Томский государственный университет систем управления и радиоэлектроники. – Электрон. текстовые дан. – Томск: [б. и.], 2012. – on-line, 94 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/training/publications/2548 (дата обращения: 04.07.2018).

12.3. Учебно-методические пособия

12.3.1. Обязательные учебно-методические пособия

1. Михальченко С.Г. Информационные технологии. Часть 1. Пограммирование на C++ [Электронный ресурс]: Руководство по организации самостоятельной работы студентов / С. Г. Михальченко; Томск. гос. ун-т сист. упр. и радиоэлектроники, Кафедра промышленной электроники – Томск: ТУСУР, 2015. – 164 с. : ил., табл., прил. – Библиогр.: с. 162. [Электронный ресурс] - Режим доступа: http://ie.tusur.ru/docs/msg/it_1.rar (дата обращения: 04.07.2018).

2. Михальченко С.Г. Информационные технологии. Часть 2. Профессиональные математические пакеты [Электронный ресурс]: Руководство по организации самостоятельной работы студентов / С. Г. Михальченко; Томск. гос. ун-т сист. упр. и радиоэлектроники, Кафедра промышленной электроники – Томск: ТУСУР, 2015. – 128 с. : ил., табл., прил. – Библиогр.: с. 97. [Электронный ресурс] - Режим доступа: http://ie.tusur.ru/docs/msg/it_2.rar (дата обращения: 04.07.2018).

3. Михальченко С. Г. Профессиональные математические пакеты. Лабораторный практикум / С. Г. Михальченко; Томский государственный университет систем управления и радиоэлектроники, Кафедра промышленной электроники – Томск: ТУСУР, 2017. – 84 с.: ил., табл., прил. – Библиогр.: с. 65. [Электронный ресурс] - Режим доступа: http://ie.tusur.ru/docs/msg/pmp.rar (дата обращения: 04.07.2018).

4. Михальченко С. Г. Профессиональные математические пакеты. Учебно-методическое пособие по проведению практических работ / С. Г. Михальченко; Томский государственный университет систем управления и радиоэлектроники, Кафедра промышленной электроники – Томск: ТУСУР, 2017. – 86 с. : ил., табл., прил. – Библиогр.: с. 72. [Электронный ресурс] - Режим доступа: http://ie.tusur.ru/docs/msg/pmp.rar (дата обращения: 04.07.2018).

12.3.2. Учебно-методические пособия для лиц с ограниченными возможностями здоровья и инвалидов

Учебно-методические материалы для самостоятельной и аудиторной работы обучающихся из числа лиц с ограниченными возможностями здоровья и инвалидов предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации.

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

- в форме электронного документа;
- в печатной форме.

12.4. Профессиональные базы данных и информационные справочные системы

1. Mathematica. Система компьютерной алгебры компании Wolfram Research. Официальный сайт компании Wolfram Research http://www.wolfram.com. Способ доступа: http://www.wolfram.com/mathematica/.

2. Maple. Программный пакет компьютерной алгебры компании Waterloo Maple Inc. Официальный сайт: http://www.maplesoft.com/. Способ доступа: http://www.maplesoft.com/products/Maple/index.aspx.

3. MatLab. Пакет математических и инженерных вычислений. Официальный сайт компа-

Согласована на портале № 36862

нии-разработчика MathWorks http://www.mathworks.com/. Способ доступа: http://www.mathworks.com/products/matlab/.

4. MathCad. Система компьютерных вычислений. Официальный сайт компании-разработчика Mathsoft http://www.mathsoft.com/, в составе PTC Community http://communities.ptc.com. Способ доступа: http://www.mathcad.com/, http://communities.ptc.com/community/mathcad

13. Материально-техническое обеспечение дисциплины и требуемое программное обеспечение

13.1. Общие требования к материально-техническому и программному обеспечению дисциплины

13.1.1. Материально-техническое и программное обеспечение для лекционных занятий

Для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используется учебная аудитория с количеством посадочных мест не менее 22-24, оборудованная доской и стандартной учебной мебелью. Имеются демонстрационное оборудование и учебно-наглядные пособия, обеспечивающие тематические иллюстрации по лекционным разделам дисциплины.

13.1.2. Материально-техническое и программное обеспечение для практических занятий

Вычислительная лаборатория

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)

634034, Томская область, г. Томск, Вершинина улица, д. 74, 201б ауд.

Описание имеющегося оборудования:

- Персональные компьютеры (16 шт.);
- Интерактивная доска – «Smart-board» DViT (1 шт.);
- Мультимедийный проектор NEC (1 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– Microsoft Visual Studio

Вычислительная лаборатория / Компьютерный класс

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)

634034, Томская область, г. Томск, Вершинина улица, д. 74, 301б ауд.

Описание имеющегося оборудования:

- Персональные компьютеры (16 шт.);
- Интерактивная доска – «Smart-board» DViT (1 шт.);
- Мультимедийный проектор NEC (1 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– Visual Studio

Лаборатория микропроцессорных устройств и систем / Лаборатория ГПО

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)

634034, Томская область, г. Томск, Вершинина улица, д. 74, 333 ауд.

Описание имеющегося оборудования:

- Лабораторные макеты (10 шт.);
- Микропроцессорный модуль «SDK-1.1» (8 шт.);

Согласована на портале № 36862

- Осциллографы (12 шт.);
- Генератор сигналов Г3-54 (2 шт.);
- Персональный компьютер (12 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– Visual Studio

13.1.3. Материально-техническое и программное обеспечение для лабораторных работ

Вычислительная лаборатория

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)

634034, Томская область, г. Томск, Вершинина улица, д. 74, 201б ауд.

Описание имеющегося оборудования:

- Персональные компьютеры (16 шт.);
- Интерактивная доска – «Smart-board» DViT (1 шт.);
- Мультимедийный проектор NEC (1 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– Microsoft Visual Studio

Вычислительная лаборатория / Компьютерный класс

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)

634034, Томская область, г. Томск, Вершинина улица, д. 74, 301б ауд.

Описание имеющегося оборудования:

- Персональные компьютеры (16 шт.);
- Интерактивная доска – «Smart-board» DViT (1 шт.);
- Мультимедийный проектор NEC (1 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– Visual Studio

Лаборатория компьютерных сетей и промышленной автоматизации / Лаборатория (ГПО) / Компьютерный класс

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, учебная аудитория для проведения занятий семинарского типа, помещение для курсового проектирования (выполнения курсовых работ)

634034, Томская область, г. Томск, Вершинина улица, д. 74, 338 ауд.

Описание имеющегося оборудования:

- Персональные компьютеры (13 шт.);
- Стенды «Промышленная электроника» Деконт-182 (7 шт.);
- Комплект имитаторов сигналов(7 шт.);
- Коммутатор 3COM SuperStackSwitch 4226T;
- Коммутатор 3COM SuperStack-3 Switch 3226;
- Коммутационный шкаф с патч-панелями;
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– Visual Studio

Согласована на портале № 36862

13.1.4. Материально-техническое и программное обеспечение для самостоятельной работы

Для самостоятельной работы используются учебные аудитории (компьютерные классы), расположенные по адресам:

- 634050, Томская область, г. Томск, Ленина проспект, д. 40, 233 ауд.;
- 634045, Томская область, г. Томск, ул. Красноармейская, д. 146, 201 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Состав оборудования:

- учебная мебель;
- компьютеры класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. - 5 шт.;
- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета.

Перечень программного обеспечения:

- Microsoft Windows;
- OpenOffice;
- Kaspersky Endpoint Security 10 для Windows;
- 7-Zip;
- Google Chrome.

13.2. Материально-техническое обеспечение дисциплины для лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины лицами с ограниченными возможностями здоровья и инвалидами осуществляется с использованием средств обучения общего и специального назначения.

При занятиях с обучающимися **с нарушениями слуха** предусмотрено использование звукоусиливающей аппаратуры, мультимедийных средств и других технических средств приема/передачи учебной информации в доступных формах, мобильной системы преподавания для обучающихся с инвалидностью, портативной индукционной системы. Учебная аудитория, в которой занимаются обучающиеся с нарушением слуха, оборудована компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

При занятиях с обучающимися **с нарушениями зрениями** предусмотрено использование в лекционных и учебных аудиториях возможности просмотра удаленных объектов (например, текста на доске или слайда на экране) при помощи видеоувеличителей для комфортного просмотра.

При занятиях с обучающимися **с нарушениями опорно-двигательного аппарата** используются альтернативные устройства ввода информации и другие технические средства приема/передачи учебной информации в доступных формах, мобильной системы обучения для людей с инвалидностью.

14. Оценочные материалы и методические рекомендации по организации изучения дисциплины

14.1. Содержание оценочных материалов и методические рекомендации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы в составе:

14.1.1. Тестовые задания

1. Что будет выведено на экран после выполнения следующего программного кода:
float y= 123.4567890; printf("x=%10.3f",y);
Варианты ответов:

- x=123,457
- y=123,4567890
- x=123,4567890
- y=123,456

Согласована на портале № 36862

2. Что будет выведено на экран после выполнения следующего программного кода:
float x= 123.4; cout << "y=" << (int)x/2;
Варианты ответов:
• y=61
• y=61.7
• x=61
• y=123.2
3. Что будет выведено на экран после выполнения следующего программного кода:
char a=65; a= a++; cout << a << endl;
если ASCII-код символа "A" равен 65?
Варианты ответов:
• B
• a=65endl
• a=66
• a=A
4. Что будет выведено на экран после выполнения следующего программного кода:
int A[10];
for(int i=0; i<< A[5] << endl;
Варианты ответов:
• 5
• 4
• i
• 10
5. Что будет выведено на экран после выполнения следующего программного кода:
int A[10];
for(int i=0; i<< *(P+2) << endl;
Варианты ответов:
• 8
• 9
• 1
• 2
6. Что будет выведено на экран после выполнения следующего программного кода:
int a= 12; int *b=&a; cout << *b+1 << endl;
если переменная а лежит по адресу 0012FF60?
Варианты ответов:
• 13
• 12
• 0012FF60
• 0012FF64
7. Что будет выведено на экран после выполнения следующего программного кода:
int a= 12; int *b=&a; cout << b+1 << endl;
если переменная а лежит по адресу 0012FEA0?
Варианты ответов:
• 0012FEA4
• 12
• 0012FEA0
• 13
8. Что будет выведено на экран после выполнения следующего программного кода:
float *x=(float*)calloc(10,sizeof(float));
for(int i=0; i, *(x+4));
Варианты ответов:
• x=6,000000
• x=6
• 4

Согласована на портале № 36862

• x=4,000000
9. Что будет выведено на экран после выполнения следующего программного кода:
double x= 12.3; double *y=&x;
y=y+1; cout << *(y-1)-1 << endl;
если переменная а лежит по адресу 0012FF6E?
Варианты ответов:
• 11,3
• 12,3
• 0012FF6E
• 0012FF6A
10. Что будет выведено на экран после выполнения следующего программного кода:
int a=12; a--;
if (a%2!=0) cout << "a%2" << endl;
else cout << "a%2-1" << endl;
Варианты ответов:
• a%2
• 1
• a%2-1
• 5
11. Какой смысл имеет приведенный ниже программный код?
int a=0; int b=0;
while (a<< b;
Варианты ответов:
• вычисление суммы целых чисел от 0 до 9
• вычисление чисел, меньших 10
• вычисление суммы целых чисел от 0 до 10
• вычисление среднего значения 10 целых чисел
12. Сколько раз выполнится тело цикла в приведенном ниже программном коде?
int a=6;
do { cout << a; a++; } while (a<10);
Варианты ответов:
• 4
• 9
• 6789
• 5
13. Что будет выведено на экран после выполнения следующего программного кода:
int a= 12; while (a<< a;
Варианты ответов:
• 25
• 24
• 12
• 13
14. Какой смысл имеет приведенный ниже программный код?
int a=12; int b=13;
if (a<< a; else cout << b;
Варианты ответов:
• вычисление наименьшего из чисел
• вычисление среднего значения двух чисел
• вычисление наибольшего из чисел
• вычисление максимума
15. Что будет выведено на экран после выполнения следующего программного кода:
int *x= new int[5]; int i=0;
for(; i<< x[i]; }
Варианты ответов:

Согласована на портале № 36862

• 01234
• 0123456789
• 012345
• 12345

16. К какому типу операторов относится запись:
do i++; while(i<100);
Варианты ответов:
• оператор цикла
• оператор выбора
• оператор ввода-вывода
• оператор присваивания

17. Что будет выведено на экран после выполнения следующего программного кода:
int y=14; f(y); cout << y; y=16;
Если функция f задана следующим образом:
void f(int x) { x++; }
Варианты ответов:
• 14
• 15
• 16
• 17

18. Что будет выведено на экран после выполнения следующего программного кода:
int y=14; f(y); cout << y; y=16;
Если функция f задана следующим образом:
void f(int &x) { x++; }
Варианты ответов:
• 15
• 17
• 16
• 14

19. Чему будет равен элемент y[2][1] матрицы y?
int y[4][4]; int *p=&y[0][0];
for(int i=0; i<< y[2][1] ;
Варианты ответов:
• 9
• 6
• 1
• 4

20. При помощи какого оператора освобождается динамическая память, если выделение памяти производилось оператором new?
int *y = new(int)
Варианты ответов:
• delete y;
• y= (int*)malloc(sizeof(int));
• free(y);
• y= (int*)calloc(1, sizeof(int));

21. Как подключить фрагменты программного кода, библиотеки и заголовочные файлы класса к своей программе?
Варианты ответов:
• используя директиву #include
• используя директиву #define
• используя файл описания *.h (header)
• используя файловую переменную FILE*

14.1.2. Экзаменационные вопросы

1. Что представляет собой программа на языке C++?

Согласована на портале № 36862

2. Что такое библиотеки и как их подключить к своей программе?

3. Как задается переход на новую строку при выводе на экран?

4. Как использовать функции потокового ввода-вывода?

5. Как в форматном спецификаторе указать, в скольких ячейках выводить число?

6. Для чего нужна строка форматной спецификации, как она выглядит?

7. Как использовать функции форматного ввода-вывода?

8. Что хранится в ячейке символьного типа (char)? Как это значение выводится на экран?

1. Что представляет собой символьная строка?

2. Как используются стандартные функции форматного и потокового ввода-вывода в работе со строками?

3. Размер символьного массива и длина строки – это одно и то же?

4. Как обозначается конец строки, для чего он нужен?

5. Какие операторы и функции служат для освобождения памяти, занятой динамическим символьным массивом?

6. Как производится присваивание (копирование) строковых переменных?

7. Может ли строка символов быть статической или динамической?

8. Как определить количество символов в массиве? В строке?

9. Можно ли обращаться к элементам строки при помощи оператора «квадратные скобки», а по технологии «адрес + смещение»?

10. Что такое адрес строки? Как сравнить две строковых переменных?

11. Какие функции служат для перевода числа в символьную строку, какие – для перевода символьной строки в число? Что возвращается, если перевод сделать не удалось?

1. Как скопировать один массив в другой?

2. Что такое динамический массив, как выделять под него память?

3. Какие операторы и функции применяются для освобождения динамической памяти?

4. Как располагаются в памяти элементы массива?

5. Что такое статический массив, какие есть два основных способа обращения к элементам массива?

6. Можно ли хранить в ячейках одномерного массива переменные разных типов?

7. Какие операторы и функции используются для выделения памяти под динамический массив?

8. Как определить адреса статического и динамического массивов, где они хранятся?

9. Являются ли эти записи идентичными для одномерного массива A: &(A[i]) или A+i?

10. Как размещается в памяти статический двумерный массив?

11. Задана матрица N×M, при такой размерности N – это число строк или столбцов?

12. Как выделяется память под матрицу, хранящуюся в виде одномерного массива?

13. Как используется вложенный цикл при работе с матрицами?

14. Как обращаться к элементам с координатами i и j матрицы, хранящейся в виде одномерного массива?

15. Чем фактически являются строки любой – статической или динамической матрицы?

16. Как передается в подпрограмму статический двумерный массив?

1. Чем отличаются операторы выбора от операторов цикла?

2. Как организовать бесконечный цикл?

3. Что такое «пошаговая трассировка программного кода»? Для чего она используется?

4. Сколько раз выполняется оператор, содержащийся в конструкции if ?

5. Как увидеть текущее значение переменной во время выполнения программы?

6. Как определить, сколько раз будет выполнен оператор, содержащийся в конструкции for?

7. Для чего используется оператор break? Что будет, если его применить в теле цикла?

8. Что такое счетчик цикла? Как его использовать в операторе for?

9. Какие значения могут принимать логические переменные? Можно ли их задавать числами? Какой объем памяти они занимают?

10. Сколько раз выполняется оператор, содержащийся в конструкции if-else ?

11. Каковы правила использования оператора множественного выбора switch? Для чего нужен блок default?

Согласована на портале № 36862

12. Чем отличается цикл с предусловием от цикла с постусловием? Приведите пример.

13. В чем особенность оператора continue? Как его применяют?

14. Что такое тернарная логическая операция? Какова семантика её применения?

15. Как используется оператор break? Что будет, если его пропустить в одном из case-блоков оператора switch?

1. Что такое указатель и для чего он используется?

2. Как узнать значение переменной, если известен только ее адрес?

3. Какие действия выполняет оператор new?

4. Имеет ли статическая переменная адрес?

5. В чем разница между динамической переменной и указателем на нее?

6. Какое значение возвращают функции malloc и calloc при выделении памяти?

7. Как получить адрес переменной?

8. Какое имя можно задать динамической переменной?

9. Что такое адрес переменной?

10. Какие пять действий происходят при выделении памяти функциями malloc, calloc?

11. Какой размер занимает переменная-указатель?

12. Что такое динамическая переменная, чем она отличается от статической?

13. В каком случае нужно пользоваться оператором delete, а в каком – функцией free( ) для освобождения памяти?

1. Как происходит вызов подпрограммы и возвращение из нее в языке C++?

2. Почему нельзя изменить переменные вызывающей функции, передаваемые в вызываемую подпрограмму по значению?

3. Что такое перегрузка функций? Составьте пример перегруженной функции.

4. Как реализована передача значений переменных вызывающей функции в тело вызываемой подпрограммы по указателю? Изменяется ли значение указателя при этом?

5. Что такое точка возврата из функции, где она находится?

6. Как передать фактические параметры в подпрограмму по ссылке? Изменятся ли их значения в вызывающей программе?

7. Для чего нужна команда return? Какой тип данных должна иметь подпрограмма, не возвращающая никакого значения?

1. Что такое текстовый файл? Опишите последовательность записи строки в файл потоковым способом.

2. Что представляют собой файловые переменные типа FILE из библиотеки как с ними работать? Какое им присваивается значение?

3. С точки зрения механизма размещения двоичных данных в памяти компьютера имеются ли особенности хранения файлов различного назначения?

4. Опишите логику последовательности чтения отдельного слова из файла для файловых переменных (FILE) из библиотеки .

5. Что представляют собой объекты файлового ввода-вывода и из библиотеки ? Как с ними работать?

6. Опишите последовательность записи числа в файл форматным способом, используя переменные типа FILE из библиотеки .

7. В чем разница использования метода getline() и оператора >> при потоковом вводе данных из файла – подпрограмм класса .

1. Возможно ли в полях структуры размещать данные различных типов?

2. Как выделяется память под динамическую переменную структурного типа, как освобождается?

3. Как размещаются в переменной структурного типа ячейки полей структуры?

4. Как осуществляется обращение к полям статической переменной структурного типа?

5. Что такое объединение (union), как размещаются поля в переменных такого типа данных?

6. Что такое указатель на структуру, как с ним работать, как обращаться к полям структурной переменной «по адресу?»

7. Можно ли указывать, сколько бит выделять под каждое поле структуры?

8. Как осуществляется обращение к полям динамической переменной структурного типа?

Согласована на портале № 36862

9. Как передается в подпрограмму статическая переменная структурного типа?

1. Как запрограммировать побитное обращение к переменным целых типов.

2. Как работает двоичная побитная операция & «и», что получится в результате вычисления 23&126?

3. Как размещаются в памяти поля переменной типа union?

4. Как работает двоичная побитная операция | «или», что получится в результате вычисления 23|112?

5. Как работает двоичная побитная операция ^ «исключающее или», что получится в результате 103^112?

6. Объединение содержит три поля unsigned short X, double Z и char Y, какой размер будет занимать переменная этого типа?

7. Как работает побитная операция «сдвиг», что получится в результате 23>>2?

8. Что такое битовые поля и как с ними работать?

14.1.3. Темы опросов на занятиях

Понятие информации. Предмет информатики. Информационные процессы. Информатизация общества. Компьютеры. Технология проектирования и отладки программ. Инструментальные средства программирования. Основные принципы создания программного обеспечения. Основы программирования на языке верхнего уровня (на примере C++). Процесс создания программного кода. Отладка программ. Трассировка программного кода.

Алгоритмические конструкции. Операторы выбора. Перечисляемый тип данных. Операторы цикла. Использование переменных логического типа. Оператор безусловного перехода. Организация диалога с пользователем.

Принципы обращения к ячейкам памяти. Адрес переменной. Указатели.

Статическое и динамическое распределение памяти. Функции динамического распределения памяти. Освобождение памяти. Операторы new и delete. Обращение к переменным "по имени" и "по адресу". Отличие указателя от динамической переменной.

Подпрограммы: процедуры и функции. Структура подпрограммы. Формальные и фактические параметры. Передача параметров в тело функции. Вызов подпрограммы. Возвращение значения, оператор return. Область видимости переменной.

Описание и вызов подпрограммы. Прототип подпрограммы. Адрес функции, указатель на подпрограмму. Область видимости переменной. Глобальные и локальные переменные. Перегрузка функций.

Массивы - хранилище однотипных данных. Одномерные (вектора) и многомерные (матрицы) массивы. Описание массива, обращение к элементам массива по индексу. Генератор случайных чисел. Работа с массивами в цикле.

Статические и динамические массивы. Функции calloc(), malloc() и free(). Адрес массива = адрес начальной ячейки. Указатель на массив. Передача массива в функцию. Динамические одномерные и двумерные массивы. Обращение к элементам массива "адрес+смещение".

Двумерные статические и динамические. Три способа размещения в памяти двумерных массивов. Освобождение памяти. Обращение к ячейкам двумерного массива.

Работа с символьными массивами. Потоковый ввод-вывод. Динамические строки.

Строки символов. Строка – массив символов. Вывод строки на экран. Последний элемент строки. Указатель на строку.

Копирование и сравнение строк. Статические и динамические строки. Функции преобразования строковых типов данных.

Файловые подсистемы ОС, хранение данных на диске. Дескриптор файла в программе. Доступ к файлу, совместный доступ, транзакции. Файловые операции (связать, открыть, закрыть, читать, писать, определить конец).

Работа с файлами при помощи потоков ввода-вывода. Библиотека . Файловые операции. Поиск в файле.

Структуры. Описание в программе, доступ к полям структуры. Размещение в памяти. Указатели на структуру. Статические и динамические переменные структурного типа. Битовые поля. Объединения. Статические и динамические переменные структурного типа.

Приемы программирования микроконтроллеров, битовые операции. Поразрядные логиче-

Согласована на портале № 36862

ские операции. Поразрядные операции сдвига. Обращение к разрядам при помощи битовых полей.

Введение в понятие "класс". Поля и методы класса. Спецификаторы доступа, использование set- и get- методов. Конструктор и деструктор.

Указатель на объект класса. Статические и динамические объекты класса. Конструктор и деструктор для статических и динамических объектов класса. Отделение интерфейса программы от реализации. Дружественные функции. Перегрузка операторов. Наследование.

Вопросы информационной безопасности. Угрозы информационных атак. Компьютерные вирусы.

14.1.4. Темы индивидуальных заданий

Технология проектирования и отладки программ. Инструментальные средства программирования. Основные принципы создания программного обеспечения. Основы программирования на языке верхнего уровня (на примере C++). Процесс создания программного кода. Отладка программ. Трассировка программного кода. Алгоритмические конструкции. Операторы выбора. Перечисляемый тип данных. Операторы цикла. Использование переменных логического типа. Оператор безусловного перехода. Организация диалога с пользователем.

Что такое текстовый файл? Опишите последовательность записи строки в файл потоковым способом. Что представляют собой файловые переменные типа FILE из библиотеки <stdio.h> как с ними работать? Какое им присваивается значение? С точки зрения механизма размещения двоичных данных в памяти компьютера имеются ли особенности хранения файлов различного назначения? Опишите логику последовательности чтения отдельного слова из файла для файловых переменных (FILE) из библиотеки <stdio.h>. Что представляют собой объекты файлового ввода-вывода <ifstream> и <ofstream> из библиотеки <iostream>? Как с ними работать? Опишите последовательность записи числа в файл форматным способом, используя переменные типа FILE из библиотеки <stdio.h>. В чем разница использования метода getline() и оператора >> при потоковом вводе данных из файла – подпрограмм класса <ifstream>.

Принципы обращения к ячейкам памяти. Адрес переменной. Указатели. Статическое и динамическое распределение памяти. Функции динамического распределения памяти. Освобождение памяти. Операторы new и delete. Обращение к переменным "по имени" и "по адресу". Отличие указателя от динамической переменной.

Описание массива, обращение к элементам массива по индексу. Генератор случайных чисел. Работа с массивами в цикле. Статические и динамические массивы. Функции calloc(), malloc() и free(). Адрес массива = адрес начальной ячейки. Указатель на массив. Передача массива в функцию. Динамические одномерные и двумерные массивы. Обращение к элементам массива "адрес+смещение". Двумерные статические и динамические. Три способа размещения в памяти двумерных массивов. Освобождение памяти. Обращение к ячейкам двумерного массива.

14.1.5. Темы контрольных работ

Контрольная работа №3. Строковые типы данных. Строка как массив. Строка как указатель. Динамические строки.

Работа с символьными массивами. Потоковый ввод-вывод.

Динамические строки.

Строка – массив символов. Вывод строки на экран.

Последний элемент строки. Указатель на строку.

Копирование и сравнение строк.

Статические и динамические строки.

Функции преобразования строковых типов данных.

Контрольная работа №4. Структурированные данные. Указатель на структуру. Динамические структуры. Объединение, битовые поля.

Структуры. Описание в программе, доступ к полям структуры.

Размещение в памяти. Указатели на структуру.

Статические и динамические переменные структурного типа.

Битовые поля. Объединения.

Приемы программирования микроконтроллеров, битовые операции.

Поразрядные логические операции. Поразрядные операции сдвига.

Обращение к разрядам при помощи битовых полей.

Согласована на портале № 36862

14.1.6. Зачёт

Понятие информации. Предмет информатики.
Информационные процессы. Информатизация общества. Компьютеры.
Технология проектирования и отладки программ.
Инструментальные средства программирования.
Основные принципы создания программного обеспечения.
Основы программирования на языке верхнего уровня (на примере C++).
Процесс создания программного кода. Отладка программ.
Трассировка программного кода.
Алгоритмические конструкции.
Операторы выбора.
Перечисляемый тип данных.
Операторы цикла.
Использование переменных логического типа.
Оператор безусловного перехода.
Организация диалога с пользователем.
Принципы обращения к ячейкам памяти. Адрес переменной.
Указатели. Адрес переменной.
Статическое и динамическое распределение памяти.
Функции динамического распределения памяти.
Освобождение памяти. Операторы new и delete.
Обращение к переменным "по имени" и "по адресу".
Отличие указателя от динамической переменной.
Подпрограммы: процедуры и функции. Структура подпрограммы.
Подпрограммы: Формальные и фактические параметры. Передача параметров в тело функции.
Вызов подпрограммы. Возвращение значения, оператор return.
Область видимости переменной.
Описание и вызов подпрограммы. Прототип подпрограммы.
Адрес функции, указатель на подпрограмму.
Область видимости переменной. Глобальные и локальные переменные.
Перегрузка функций.
Массивы - хранилище однотипных данных.
Одномерные (вектора) и многомерные (матрицы) массивы.
Описание массива, обращение к элементам массива по индексу.
Генератор случайных чисел.
Работа с массивами в цикле.
Статические и динамические массивы.
Функции calloc(), malloc() и free().
Адрес массива = адрес начальной ячейки.
Указатель на массив.
Передача массива в функцию.
Динамические одномерные и двумерные массивы.
Обращение к элементам массива по индексу и "адрес+смещение".
Двумерные статические и динамические массивы.
Три способа размещения в памяти двумерных массивов. Освобождение памяти.
Обращение к ячейкам двумерного массива.
Работа с символьными массивами. Потоковый ввод-вывод.
Динамические строки.
Строка – массив символов. Вывод строки на экран.
Последний элемент строки. Указатель на строку.
Копирование и сравнение строк.
Статические и динамические строки.
Функции преобразования строковых типов данных.

Согласована на портале № 36862

14.1.7. Темы лабораторных работ

Научиться создавать подпрограммы и вызывать их из тела основной функции. Освоить применение оператора return, возвращающего значение функции. Научиться передавать параметры в функцию и корректно возвращать значения – результаты вычислений из функции. Освоить передачу параметров в функцию по значению, по ссылке и по указателю. Научиться создавать прототипы функций, освоить перегрузку функций. Изучить функции библиотеки math.h. Научиться пользоваться средствами пошаговой трассировки кода и просмотра текущих значений переменных.

Контрольная работа №1. Программа. Операторы. Ввод-вывод, Переменные. Адреса.

Научиться работать с файловыми переменными посредстваом подпрограмм библиотек fstream.h и stdio.h. Закрепить навыки работы со строками. Научиться открывать файлы для записи и чтения, создавать и уничтожать файлы, определять конец файла.

Операционная система. Файловые таблицы. Элементы информационной безопасности.

Освоить умение работать с разрядами в байте. Научиться изменять произвольный бит в байте, не меняя остальных. Выводить на экран двоичный код переменной произвольного типа. Повысить понимание преобразований двоичного формата. Применять операции поразрядного сдвига и поразрядные логические операции, а также при помощи структур объединение (union) и битовые поля.

Контрольная работа №4. Структурированные данные. Указатель на структуру. Динамические структуры. Объединение, битовые поля.

Научиться создавать массивы, выделять память под элементы массива и инициировать их значениями. Научиться обращаться к элементам массивов как при помощи оператора имя[индекс], так и посредством методики *(адрес+смещение). Освоить понятия: адрес массива, адрес элемента массива, смещение, индекс. Освоить пошаговую трассировку программы с массивами, научиться отражать в окне watch элементы массива и их адреса. Научиться передавать массив в функцию.

Освоить способ динамического захвата и освобождения памяти под двумерные массивы данных (матрицы) – при помощи оператора new и посредством функций malloc( ) и calloc( ). Повторить понятия, операции и закрепить умения и навыки матричной алгебры. Уяснить практическую разницу и сходство в языке С++ понятий «массив» и «указатель». Повысить навыки отладки программного кода на С++, трассировки программы и просмотра значений переменных в окне Watch.

Контрольная работа №2. Массив. Обращение к элементам массива. Указатель на массив. Адрес + смещение. Динамические массивы.

Научиться работать со строками. Освоить способ динамического захвата и освобождения памяти под символьные массивы. Уяснить разницу между понятиями «размер массива символов» и «длина строки» при помощи функций sizeof и strlen. Освоить функции библиотеки «string.h». Научится переводить числовые данные в строковые и обратно.

Операционная система. Файловые таблицы. Обращение к оборудованию ПК. BIOS. Setup BIOS. Драйверы.

14.1.8. Методические рекомендации

Дисциплина является обеспечивающей, формируемые компетенции требуются для освоения предметов, непосредственно относящихся к электронике и наноэлектронике, силовой электронике (power electronics) и промышленной электронике (industrial electronics). В связи с этим при проведении всех видов занятий и форм контроля необходимо заострять внимание обучающихся на практическом применении получаемых компетенций в предметной области будущей профессиональной деятельности.

14.2. Требования к оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусмотрены дополнительные оценочные материалы, перечень которых указан в таблице 14.

Таблица 14 – Дополнительные материалы оценивания для лиц с ограниченными возможностями здоровья и инвалидов

Категории обучающихся	Виды дополнительных оценочных материалов	Формы контроля и оценки результатов обучения

Согласована на портале № 36862

С нарушениями слуха	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы	Преимущественно письменная проверка
С нарушениями зрения	Собеседование по вопросам к зачету, опрос по терминам	Преимущественно устная проверка (индивидуально)
С нарушениями опорно-двигательного аппарата	Решение дистанционных тестов, контрольные работы, письменные самостоятельные работы, вопросы к зачету	Преимущественно дистанционными методами
С ограничениями по общемедицинским показаниям	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы, устные ответы	Преимущественно проверка методами исходя из состояния обучающегося на момент проверки

14.3. Методические рекомендации по оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной форме;
- в печатной форме с увеличенным шрифтом;
- в форме электронного документа;
- методом чтения ассистентом задания вслух;
- предоставление задания с использованием сурдоперевода.

Лицам с ограниченными возможностями здоровья и инвалидам увеличивается время на подготовку ответов на контрольные вопросы. Для таких обучающихся предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге;
- набор ответов на компьютере;
- набор ответов с использованием услуг ассистента;
- представление ответов устно.

Процедура оценивания результатов обучения лиц с ограниченными возможностями здоровья и инвалидов по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

- в форме электронного документа;
- в печатной форме.

При необходимости для лиц с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения может проводиться в несколько этапов.

Согласована на портале № 36862