

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

ПРИЕМ 2019 г.

ФОРМА ОБУЧЕНИЯ очная

Программирование

Направление подготовки/ специальность	09.03.01 Информатика и вычислительная техника		
Образовательная программа (направленность (профиль))	Программирование вычислительных и телекоммуникационных систем		
Специализация	Информационно-коммуникационные технологии		
Уровень образования	высшее образование – бакалавриат		
Курс	1, 2	семестр	3, 4
Трудоемкость в кредитах (зачетных единицах)	6		

Заведующий кафедрой - руководитель отделения на правах кафедры		Шерстнев В.С.
Руководитель ООП		Погребной А.В.
Преподаватель		Рейзлин В.И.

2020 г.

Роль дисциплины «Программирование» в формировании компетенций выпускника:

Элемент образовательной программы (дисциплина, практика, ГИА)	Семестр	Код компетенции	Наименование компетенции	Индикаторы достижения компетенций		Составляющие результатов освоения (дескрипторы компетенций)	
				Код индикатора	Наименование индикатора достижения	Код	Наименование
Программирование	2, 3	УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	И.УК(У)-1.2	Осуществляет поиск, выделяет и ранжирует информацию на основе системного подхода и методов познания для решения задач по различным типам запросов	УК(У)-1.2В2	Имеет практический опыт работы с информационными источниками, опыт научного поиска, создания научных текстов.
						УК(У)-1.2У2	Умеет соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности.
						УК(У)-1.2З2	Знает принципы сбора, отбора и обобщения информации.
		ОПК(У)-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения. Способен разрабатывать требования и проектировать программное обеспечение	И.ОПК(У)-8.2	Демонстрирует способность написания программного кода с использованием языков программирования, определения и манипулирования данными	ОПК(У)-8.2В1	Владеет навыками создания программного кода в соответствии с техническим заданием (готовыми спецификациями)
						ОПК(У)-8.2У1	Умеет применять выбранные языки программирования для написания программного кода
						ОПК(У)-8.2З1	Знает синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования

1. Показатели и методы оценивания

Планируемые результаты обучения по дисциплине	Код индикатора	Наименование	Методы оценивания
---	----------------	--------------	-------------------

Код	Наименование	достижения контролируемой компетенции (или ее части)	раздела дисциплины	(оценочные мероприятия)
РД1	Выполнять самостоятельное изучение отдельных тем дисциплины и решения типовых задач, используя поиск информации в библиографических источниках и сети интернет	И.УК(У)-1.2	Разделы 1 - 5	Защита отчета по лабораторной работе, итоговое независимое тестирование ЦОКО
РД2	Способен разрабатывать и отлаживать эффективные алгоритмы и программы с использованием структурного программирования на языке C++ в современных средах программирования	И.ОПК(У)-8.2	Разделы 1-3	Защита отчета по лабораторной работе, тестирование в электронной образовательной среде Moodle, итоговое независимое тестирование ЦОКО
РД3	Использует современные методы объектно-ориентированного программирования, шаблоны, библиотеки, типовые структуры данных при разработке и кодировании алгоритмов разного уровня сложности	И.ПК(У)-1.1	Разделы 4, 5	Защита отчета по лабораторной работе, итоговое независимое тестирование ЦОКО

2. Шкала оценивания

Порядок организации оценивания результатов обучения в университете регламентируется отдельным локальным нормативным актом – «Система оценивания результатов обучения в Томском политехническом университете (Система оценивания)» (в действующей редакции). Используется балльно-рейтинговая система оценивания результатов обучения. Итоговая оценка (традиционная и литерная) по видам учебной деятельности (изучение дисциплин, УИРС, НИРС, курсовое проектирование, практики) определяется суммой баллов по результатам текущего контроля и промежуточной аттестации (итоговая рейтинговая оценка - максимум 100 баллов).

Распределение основных и дополнительных баллов за оценочные мероприятия текущего контроля и промежуточной аттестации устанавливается календарным рейтинг-планом дисциплины.

Рекомендуемая шкала для отдельных оценочных мероприятий входного и текущего контроля

% выполнения задания	Соответствие традиционной оценке	Определение оценки
90%÷100%	«Отлично»	Отличное понимание предмета, всесторонние знания, отличные умения и владение опытом практической деятельности, необходимые результаты обучения сформированы, их качество оценено количеством баллов, близким к максимальному
70% - 89%	«Хорошо»	Достаточно полное понимание предмета, хорошие знания, умения и опыт практической деятельности, необходимые результаты обучения сформированы, качество ни одного из них не оценено минимальным количеством баллов
55% - 69%	«Удовл.»	Приемлемое понимание предмета, удовлетворительные знания, умения и опыт практической деятельности, необходимые результаты обучения сформированы, качество некоторых из них оценено минимальным количеством баллов
0% - 54%	«Неудовл.»	Результаты обучения не соответствуют минимально достаточным требованиям

3. Перечень типовых заданий

	Оценочные мероприятия	Примеры типовых контрольных заданий
1.	Тестирование	Вопросы: 1. Выберите правильный набор символов, входящий в алфавит языка C++ 1) A } [] () b + - = ^ * . ф , / % \ ; : < = > ~ & _ " 2) " { } , [] + () = ? ^ * . / \ ; ' : < = > ~ # & ! _ " 3) { } [] () + - = ? ^ * . \$, / % \ ; : < = > ~ # & _ " 4) [] () ? + - " { } = ? . , * ^ / % \ ; : @ < = > ~ # ! 2. Определите значение переменных a, b, c после выполнения следующих операций: <pre>int a=0, b=1, c; c=a++; c=++a; c=++b;</pre> В ответ запишите 3 числа через пробел (a b c) 3. Определите значения переменных d1, d2, d3 после выполнения действий: <pre>double d1,d2,d3; d1 = 2.56; d2 = (int)d1 + 1.5; d3 = (int)(d1 + 1.5);</pre> 4. Запишите значение переменной u после выполнения действий: <pre>unsigned u2, u; int u1 = -1; u2 = -1; u=0; if (u1 != u2) u++;</pre> 5. void main() <pre>{int i=3; {int c = i; int m=3*c; int i = 0; cout <<m <<"\n";</pre>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<pre> } cout << i << "\n"; system("pause"); } </pre> При выполнении этого фрагмента кода на консоли появятся числа _____ <i>Запишите два числа через пробел</i> 6. int *a; int b; a = &b; b = 7; *a++; cout << b; system ("pause"); После выполнения этого фрагмента программы на консоли появится число _____ 7. char x[] = "Hello world"; char* p = x; p += 4; cout << *p; system ("pause"); После выполнения этого фрагмента программы на консоли появится 1) o 2) o world 3) Hello world 4) ;5\$%%#0 8. Прототип функции задает: 1) тип функции, включая количество и тип аргументов и тип результата 2) возможность выполнения этой функции из программ на других языках программирования 3) имя функции и минимальное количество ее параметров 4) стандартная функция форматированного вывода, с помощью которой осуществляется вывод данных в стандартный поток 9. Укажите верное утверждение 1) Две функции с одним и тем же числом, порядком и типом аргументов могут быть перегружены, если обе не имеют аргументов по умолчанию 2) Перегруженные функции не должны иметь параметры по умолчанию

	Оценочные мероприятия	Примеры типовых контрольных заданий
		3) Перегруженные функции должны иметь параметры по умолчанию, начинающиеся слева в списке аргументов 4) Функция может быть перегружена более, чем один раз 10. Отметьте, какому определению функции может соответствовать вызов func(5.98) 1) int func(double x = 0, double y); 2) void func(double x); 3) double func(double x, int y = 12, int z = 5); 4) void func(float arg1, float arg2 = 0); 5) float func(float g, float f); 11. Укажите, какая из записей соответствует обращению к члену m_arg класса AC в определении метода этого же класса 1) this.m_arg.AC 2) m_arg 3) AC.this.m_arg 4) AC->m_arg 12. Абстрактный класс – это класс, в котором есть 1) виртуальный конструктор 2) виртуальный деструктор 3) хотя бы одна чисто виртуальная функция 4) абстрактный класс определяет неявно некоторое множество объектов, задавая применимые к ним функции 13. Конструктор класса – это метод, который вызывается при создании экземпляра класса с целью 1) загрузки всех методов класса в память 2) задания свойств объекта

	Оценочные мероприятия	Примеры типовых контрольных заданий
		3) выделения памяти под создаваемый экземпляр класса и инициализации этой памяти 4) выгрузки методов класса из памяти 14. Известно, что в классе A определен один публичный конструктор A(int); Выберите из предложенных выражений компилируемые 1) A *a(4); 2) A a; 3) A *a = new A(4); 4) A a(4); 15. Функция с тем же именем, что и класс, которому предшествует символ тильда (~) называется _____ этого класса. 16. Переопределение операции сложения приведет к 1) ее вызову при выполнении операции ++ с объектом класса 2) ее вызову при выполнении операции сложения с объектом класса 3) преобразованию целых чисел к объекту данного класса при выполнении сложения 4) возможному преобразованию объектов других классов к данному при выполнении операции сложения 17. Имеется шаблон функции: <pre>template T func1(T a, T b) {if(a) a=a%b; return a; }</pre> Следующий код: <pre>double a=5, b=6, c; c=func1(a, b);</pre>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		1) полностью верен 2) неверен, ошибка связана с типом операндов в операторе if 3) неверен, ошибка связана с типом операндов в операции % 18. В классе A определены методы A(int x); operator int(); operator++(int); Укажите, какие из них будут вызваны в следующем фрагменте: A b; static_cast (b + 1); 1) operator int(), конструктор 2) operator++(int), operator int(), конструктор 3) только конструктор 4) только operator int() 5) operator++(int), конструктор 19. При определении класса-шаблона 1) он должен быть включен в общий контекст (namespace) STL 2) он должен быть наследован от класса Template 3) он должен быть отмечен ключевым словом template 4) он должен включать в себя, по крайней мере, один параметр-тип 20. Возможность создания нескольких функций с одинаковыми именами – это _____ функций 1) наследование 2) инкапсуляция 3) полиморфизм 4) переопределение

	Оценочные мероприятия	Примеры типовых контрольных заданий
2.	Защита лабораторной работы	Вопросы: Что такое интегрированная среда разработки. Что такое хедер. Директивы препроцессора. Выражение присваивания. Арифметические операции с целыми и плавающими переменными. Логические операции, операции автоувеличения и автоуменьшения, тернарная операция. Составной оператор. Условный оператор. Оператор switch - case. Оператор безусловного перехода, break, continue. Операторы цикла. Оператор безусловного перехода, break, continue. Указатели. Указатели и массивы. Адресная арифметика. Символьные массивы и строки. Указатели и многомерные массивы. Операции для работы с динамической памятью. Объявления и определения. Область существования имени. Область видимости имён. Классы памяти. Функции. Передача аргументов. Указатели на функции. Ссылки. Передача аргументов в функции по ссылке. Функции. Аргументы по умолчанию и переопределение функций. Что такое абстрактный тип данных? Приведите примеры абстрактных типов данных. Каковы синтаксис/семантика “операции-функции”? Как можно вызвать операцию-функцию? Нужно ли перегружать операцию присваивания относительно определенного пользователем типа данных, например, класса? Почему? Можно ли изменить приоритет перегруженной операции? Можно ли изменить количество операндов перегруженной операции? Можно ли изменить ассоциативность перегруженной операции? Можно ли, используя дружественную функцию, перегрузить оператор присваивания? Все ли операторы языка C++ могут быть перегружены? В чем смысл использования шаблонов? Каковы синтаксис/семантика шаблонов функций? Каковы синтаксис/семантика шаблонов классов? Определите шаблон класса “вектор” – одномерный массив. Что такое параметры шаблона функции? Перечислите основные свойства параметров шаблона функции.

	Оценочные мероприятия	Примеры типовых контрольных заданий
		Как записывать параметр шаблона? Можно ли перегружать параметризованные функции? Перечислите основные свойства параметризованных классов. Может ли быть пустым список параметров шаблона? Объясните. Как вызвать параметризованную функцию без параметров? Все ли компонентные функции параметризованного класса являются параметризованными? Являются ли дружественные функции, описанные в параметризованном классе, параметризованными? Могут ли шаблоны классов содержать виртуальные компонентные функции?
3.	Задания к лабораторным работам	Первый семестр Лабораторная работа №1 Арифметические операции. Условный оператор Разработать программу для вычисления заданных выражений и вывода полученных результатов. Соответствующие исходные данные ввести с клавиатуры. Вариант 1 $a = \ln(y^{-\sqrt{ x }}) \cdot (\sin(x) + e^{(x+y)})$ $b = \begin{cases} \ln(x/y) + (x^2 + y)^3, & x/y > 0 \\ \ln x/y + (x^2 + y)^3, & x/y < 0 \\ (x^2 + y)^3, & y \neq 0, x = 0 \\ 0, & y = 0 \end{cases}$ Лабораторная работа №2 Операторы цикла. Табулирование функции Вычислить и вывести на экран таблицу функции в интервале [a, b] с шагом h. Вариант 14

Оценочные мероприятия		Примеры типовых контрольных заданий																																			
		$y = \begin{cases} \frac{x^3}{3+x}, & x < -3 \\ 0, & x = 3 \\ 9x - \frac{27}{7} \ln(3+x), & \text{иначе} \end{cases}$ $x \in [-4, -2], h = 0.1$ Лабораторная работа №3 Побитовые операции Для всех вариантов (если речь идет о <i>последовательностях</i>) результаты выдать в виде: <table><tr><td>Член 1 посл.</td><td>Восьм. код 1</td><td>Член 1 изм. посл.</td><td>Восьм. код 1 изм. посл.</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>Член 8 посл.</td><td>Восьм. код. 8</td><td>Член 8 изм. посл.</td><td>Восьм. код 8 изм. посл.</td></tr></table> 1. Ввести последовательность из 8 символов. В каждом из символов в их двоичном представлении заменить: для нечетных (по порядку) символов 3-й бит единицей; для четных символов – 4-й бит нулем. Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды. Лабораторная работа №4 Рекуррентные последовательности. Операторы цикла 4.7. Вычислить произведение				Член 1 посл.	Восьм. код 1	Член 1 изм. посл.	Восьм. код 1 изм. посл.	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	Член 8 посл.	Восьм. код. 8	Член 8 изм. посл.	Восьм. код 8 изм. посл.
Член 1 посл.	Восьм. код 1	Член 1 изм. посл.	Восьм. код 1 изм. посл.																																		
...	...	...	...																																		
...	...	...	...																																		
...	...	...	...																																		
...	...	...	...																																		
...	...	...	...																																		
...	...	...	...																																		
Член 8 посл.	Восьм. код. 8	Член 8 изм. посл.	Восьм. код 8 изм. посл.																																		

	Оценочные мероприятия	Примеры типовых контрольных заданий
		$P = \prod_{i=1}^{15} U_i,$ где последовательность $\{U_i\}$ задается так: $U_1 = 0.4;$ $U_2 = 0.5;$ $U_3 = 0.93;$ $U_i = \sin(U_{i-1}) + \cos(U_{i-3})$ $i = 4, 5, \dots, 15.$ Массивом не пользоваться. Лабораторная работа №5 Одномерные массивы 5.1. Даны действительные числа $a_1, \dots, a_{15}$. Получить $t = \frac{1}{15} \sum_{i=1}^{15} a_i, \quad s = \sqrt{\frac{\sum_{i=1}^{15} (a_i - t)^2}{14}}.$ Лабораторная работа №6 Двумерные массивы 6.1. Задана матрица $Z(5,4)$. Найти в каждой строке, если там есть отрицательный элемент, среднее арифметическое всех элементов, исключая нулевые и записать эти значения в массив В. Вывести исходную матрицу Z и массив В.

	Оценочные мероприятия	Примеры типовых контрольных заданий
		Лабораторная работа №7 Работа с динамической памятью (операции <i>new</i> и <i>delete</i>) 7.1. Получить квадратную матрицу порядка <i>n</i>: $\begin{bmatrix} n & 0 & \dots\dots\dots & 0 \\ n-1 & n & 0 & \dots\dots & 0 \\ n-2 & n-1 & n & 0 & \dots & 0 \\ & & \dots & & & \\ 1 & 2 & 3 & \dots\dots & n \end{bmatrix}$ Матрица размещается в памяти динамически с помощью операции <i>new</i>, значение <i>n</i> вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу. Лабораторная работа №8 Работа с файлами, символьными массивами и строками 8.1. Прочитать в символьный массив из файла строку символов. Из этой строки получить символы между первым и вторым двоеточием. Новую строку не создавать. Вывести исходную и преобразованную строки. Лабораторная работа №9 Функции. Передача аргументов. Аргументы по умолчанию Задание: Вычислить интеграл (для нечетных вариантов – методом средних прямоугольников, для четных – методом Симпсона). Для этого составить функцию, реализующую вычислительный метод. <i>Подынтегральную функцию передать в качестве параметра</i> (указатель на функцию). Предусмотреть использование <i>одного параметра по умолчанию</i> (число разбиений интервала интегрирования). Интеграл вычислить: 1. С умалчиваемым значением параметра; 2. С другим значением этого параметра; 3. По формуле Ньютона-Лейбница.

	Оценочные мероприятия	Примеры типовых контрольных заданий
		Сравнить полученные значения. 9.1. $\int_1^{3.5} \frac{\ln x}{x\sqrt{1+\ln x}} dx$ Лабораторная работа №10 Системы счисления 1. Дано натуральное число n. Определить, сколько различных цифр встречается в его десятичной записи. Второй семестр Лабораторная работа №1 Файлы и структуры 1. Читая данные из файла и используя массив структур, решить следующую задачу. Дана ведомость абитуриентов, сдавших вступительные экзамены в университет. В каждой строке данной ведомости записана фамилия абитуриента, его постоянное местожительство и полученные им оценки по отдельным дисциплинам (например, физике, математике, литературе). Необходимо определить количество абитуриентов, проживающих в выбранном городе и сдавших экзамены со средним баллом не ниже 4, распечатать их фамилии в алфавитном порядке. Лабораторная работа №2 Классы. Конструкторы и деструкторы 1. Необходимо реализовать класс с двумя полями, имеющими имена first и second. Обязательно должны присутствовать: • Для инициализации начальных значений использовать конструктор; • Обязательно использовать деструктор для удаления объектов; • Вывод на экран Display();

	Оценочные мероприятия	Примеры типовых контрольных заданий
		• Для проверки программы использовать несколько случайных наборов данных для переменных first и second. Вывод результата на экран консоли. Поле first – дробное число; поле second – целое число, показатель степени. Реализовать метод power() – возведение числа first в степень second. Метод должен правильно работать при любых допустимых значениях first и second. Лабораторная работа №3 Наследование 1. Создать базовый класс Car (машина), характеризующийся торговой маркой (строка), числом цилиндров, мощностью. Определить функции переназначения марки и изменения мощности. Создать производный класс Loggy (грузовик), характеризующийся также грузоподъемностью кузова. Определить функцию изменения грузоподъемности. Для демонстрации работы программы необходимо создать объект производного класса, задать все поля данного объекта и продемонстрировать использование всех доступных функций базового и производного классов, например, в консоли. Лабораторная работа №4 Переопределение стандартных операций 1. Комплексное число представляется парой действительных чисел (a, b), где a – действительная часть, b – мнимая часть. Реализовать класс Complex для работы с комплексными числами. Необходимо переопределить следующие операции: сложения +, $(a, b) + (c, d) = (a + b, c + d)$; вычитания -, $(a, b) - (c, d) = (a - b, c - d)$; умножения *, $(a, b) * (c, d) = (ac - bd, ad + bc)$; деления /, $(a, b) / (c, d) = (ac + bd, bc - ad) / (c^2 + d^2)$; сравнения ==, $(a, b) = (c, d)$, если $(a = c)$ и $(b = d)$; Для демонстрации работы программы необходимо создать объект класса, задать все поля данного объекта и продемонстрировать использование всех доступных функций класса, например, в консоли.

	Оценочные мероприятия	Примеры типовых контрольных заданий
		Лабораторная работа №5 Шаблоны классов 1. Создать шаблон класса для работы с массивом произвольного типа данных. Шаблон должен включать: • указатель, хранящий адрес размещения массива в динамической памяти; • целочисленную переменную, показывающую количество занятых элементов массива; • конструктор без параметров с параметрами по умолчанию; • конструктор копирования; • метод «обработка массива» (на начальном этапе метод выводит на консоль какое-либо сообщение); • деструктор и другие необходимые для работы методы (например, переопределение операций индексирования и присваивания, методы ввода / вывода и т.д.). Задание: Дана строка латинских символов. Преобразовать строку, разместив символы в алфавитном порядке, разделяя их пробелами. Для созданного программного обеспечения провести тестирование на нескольких случайных наборах данных. Лабораторная работа №6 Динамические структуры данных. Списки, стеки, двоичные деревья 2. Для решения задачи необходимо подобрать подходящую для этой задачи динамическую структуру данных. Дан текст. Среди литер этого текста особую роль играет знак #, появление которого в тексте означает отмену предыдущей литеры текста; k знаков # подряд отменяют k предыдущих литер (если такие есть). Напечатать данный текст, исправленный с учетом такой роли знака # (например, текст ХЭ#Е##HELO#LO должен быть напечатан в виде HELLO).

4. Методические указания по процедуре оценивания

5.

Проводятся методические материалы (процедуры проведения) ко всем оценочным мероприятиям:

	Оценочные мероприятия	Процедура проведения оценочного мероприятия и необходимые методические указания
1.	Тестирование	Тестирование (независимое) проводится Центром обеспечения качества обучения ТПУ в соответствии с утвержденной программой и спецификацией
2.	Защита лабораторной работы	Студенты представляют письменный отчет по лабораторной работе. При верном выполнении работы и после опроса по теме работы выставляются баллы в соответствии с рейтинг-планом качеством ответв.