

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

ПРИЕМ 2016 г.

ФОРМА ОБУЧЕНИЯ заочная

Программирование на C++

Направление подготовки/ специальность	09.03.01 Информатика и вычислительная техника		
Образовательная программа (направленность (профиль))	Информатика и вычислительная техника		
Специализация	Вычислительные машины, комплексы, системы и сети		
Уровень образования	высшее образование - бакалавриат		
Курс	1/2	семестр	2/3
Трудоемкость в кредитах (зачетных единицах)	6		

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

2020 г.

1. Роль дисциплины «Программирование на С++» в формировании компетенций выпускника:

Элемент образовательной программы (дисциплина, практика, ГИА)	Семестр	Код компетенции	Наименование компетенции	Результаты освоения ООП	Составляющие результатов освоения (дескрипторы компетенций)	
					Код	Наименование
Программирование на С++	3	ОПК(У)-2	Способен осваивать методики использования программных средств для решения практических задач	Р4	ОПК(У)-2В1	Владеет опытом применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.
					ОПК(У)-2У1	Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.
					ОПК(У)-231	Знает современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.
		ПК(У)-2	Способен разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования	Р4	ПК(У)-2В10	Владеет навыками создания программного кода в соответствии с техническим заданием (готовыми спецификациями)
					ПК(У)-2У10	Умеет применять выбранные языки программирования для написания программного кода
					ПК(У)-2310	Знает синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования

2. Показатели и методы оценивания

Планируемые результаты обучения по дисциплине		Код контролируемой компетенции (или ее части)	Наименование раздела дисциплины	Методы оценивания (оценочные мероприятия)
Код	Наименование			
РД1	Готовность выпускника разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования и в современных средах	ОПК(У)-2	Разделы 1 - 5	Защита отчета по лабораторной работе, итоговое независимое тестирование ЦОКО
РД2	Готовность выпускника использовать современные методы структурного программирования при кодировании программных систем разного уровня сложности	ОПК(У)-2	Разделы 1-3	Защита отчета по лабораторной работе, тестирование в электронной образовательной среде Moodle, итоговое независимое тестирование ЦОКО
РД3	Готовность выпускника использовать современные методы объектно-ориентированного программирования, парадигму шаблонов и структур данных при кодировании программных систем разного уровня сложности	ПК(У)-2	Разделы 4, 5	Защита отчета по лабораторной работе, итоговое независимое тестирование ЦОКО

3. 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

Шкала для оценочных мероприятий экзамена

% выполнения заданий экзамена	Экзамен, балл	Соответствие традиционной оценке	Определение оценки
90%÷100%	18 ÷ 20	«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%	14 ÷ 17	«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%	11 ÷ 13	«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%	0 ÷ 10	«Неудовл.»	Результаты обучения не соответствуют минимально достаточным требованиям

Шкала для оценочных мероприятий и дифференцированного зачета / зачета

Степень сформированности результатов обучения	Балл	Соответствие традиционной оценке	Определение оценки
90% ÷ 100%	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%	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%	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
55% ÷ 100%	55 ÷ 100	«Зачтено»	Результаты обучения соответствуют минимально достаточным требованиям
0% ÷ 54%	0 ÷ 54	«Неудовл.»/ «Не зачтено»	Результаты обучения не соответствуют минимально достаточным требованиям

4. Перечень типовых заданий

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

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<pre> unsigned u2, u; int u1 = -1; u2 = -1; u=0; if (u1 != u2) u++; 5. void main() {int i=3; {int c = i; int m=3*c; int i = 0; cout <<m <<"\n"; } cout << i <<"\n"; system("pause"); } 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++; </pre>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<pre>cout << b; system ("pause");</pre> <p>После выполнения этого фрагмента программы на консоли появится число _____</p> <p>7. <code>char x[] = "Hello world"; char* p = x; p += 4; cout << *p;</code> <code>system ("pause");</code></p> <p>После выполнения этого фрагмента программы на консоли появится</p> <p>1) o</p> <p>2) o world</p> <p>3) Hello world</p> <p>4) ;5\$%%#0</p> <p>8. Прототип функции задает:</p> <ol style="list-style-type: none"> 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 <p>9. Укажите верное утверждение</p> <ol style="list-style-type: none"> 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
		<p>3) Перегруженные функции должны иметь параметры по умолчанию, начинающиеся слева в списке аргументов</p> <p>4) Функция может быть перегружена более, чем один раз</p> <p>10. Отметьте, какому определению функции может соответствовать вызов func(5.98)</p> <p>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);</p> <p>11. Укажите, какая из записей соответствует обращению к члену m_arg класса AC в определении метода этого же класса</p> <p>1) this.m_arg.AC 2) m_arg 3) AC.this.m_arg 4) AC->m_arg</p> <p>12. Абстрактный класс – это класс, в котором есть</p> <p>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</p> <p>13. Конструктор класса – это метод, который вызывается при создании экземпляра класса с целью</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>1) загрузки всех методов класса в память</p> <p>2) задания свойств объекта</p> <p>3) выделения памяти под создаваемый экземпляр класса и инициализации этой памяти</p> <p>4) выгрузки методов класса из памяти</p> <p>14. Известно, что в классе A определен один публичный конструктор A(int); Выберите из предложенных выражений компилируемые</p> <p>1) A *a(4);</p> <p>2) A a;</p> <p>3) A *a = new A(4);</p> <p>4) A a(4);</p> <p>15. Функция с тем же именем, что и класс, которому предшествует символ тильда (~) называется _____ этого класса.</p> <p>16. Переопределение операции сложения приведет к</p> <p>1) ее вызову при выполнении операции ++ с объектом класса</p> <p>2) ее вызову при выполнении операции сложения с объектом класса</p> <p>3) преобразованию целых чисел к объекту данного класса при выполнении сложения</p> <p>4) возможному преобразованию объектов других классов к данному при выполнении операции сложения</p> <p>17. Имеется шаблон функции:</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<pre>template T func1(T a, T b) { if(a) a=a%b; return a; } Следующий код: double a=5, b=6, c; c=func1(a, b); 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);</pre>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>1) operator int(), конструктор</p> <p>2) operator++(int), operator int(), конструктор</p> <p>3) только конструктор</p> <p>4) только operator int()</p> <p>5) operator++(int), конструктор</p> <p>19. При определении класса-шаблона</p> <p>1) он должен быть включен в общий контекст (namespace) STL</p> <p>2) он должен быть наследован от класса Template</p> <p>3) он должен быть отмечен ключевым словом template</p> <p>4) он должен включать в себя, по крайней мере, один параметр-тип</p> <p>20. Возможность создания нескольких функций с одинаковыми именами – это _____ функций</p> <p>1) наследование</p> <p>2) инкапсуляция</p> <p>3) полиморфизм</p> <p>4) <u>переопределение</u></p>
2.	Защита лабораторной работы	<p>Вопросы:</p> <p>Что такое интегрированная среда разработки.</p> <p>Что такое хедер.</p> <p>Директивы препроцессора.</p> <p>Выражение присваивания. Арифметические операции с целыми и плавающими переменными.</p> <p>Логические операции, операции автоувеличения и автоуменьшения, тернарная операция.</p> <p>Составной оператор. Условный оператор.</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>Оператор switch - case. Оператор безусловного перехода, break, continue.</p> <p>Операторы цикла. Оператор безусловного перехода, break, continue.</p> <p>Указатели. Указатели и массивы. Адресная арифметика.</p> <p>Символьные массивы и строки. Указатели и многомерные массивы.</p> <p>Операции для работы с динамической памятью.</p> <p>Объявления и определения. Область существования имени.</p> <p>Область видимости имён. Классы памяти.</p> <p>Функции. Передача аргументов. Указатели на функции.</p> <p>Ссылки. Передача аргументов в функции по ссылке.</p> <p>Функции. Аргументы по умолчанию и переопределение функций.</p> <p>Что такое абстрактный тип данных?</p> <p>Приведите примеры абстрактных типов данных.</p> <p>Каковы синтаксис/семантика “операции-функции”?</p> <p>Как можно вызвать операцию-функцию?</p> <p>Нужно ли перегружать операцию присваивания относительно определенного пользователем типа данных, например, класса? Почему?</p> <p>Можно ли изменить приоритет перегруженной операции?</p> <p>Можно ли изменить количество операндов перегруженной операции?</p> <p>Можно ли изменить ассоциативность перегруженной операции?</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>Можно ли, используя дружественную функцию, перегрузить оператор присваивания?</p> <p>Все ли операторы языка C++ могут быть перегружены?</p> <p>В чем смысл использования шаблонов?</p> <p>Каковы синтаксис/семантика шаблонов функций?</p> <p>Каковы синтаксис/семантика шаблонов классов?</p> <p>Определите шаблон класса “вектор” – одномерный массив.</p> <p>Что такое параметры шаблона функции?</p> <p>Перечислите основные свойства параметров шаблона функции.</p> <p>Как записывать параметр шаблона?</p> <p>Можно ли перегружать параметризованные функции?</p> <p>Перечислите основные свойства параметризованных классов.</p> <p>Может ли быть пустым список параметров шаблона? Объясните.</p> <p>Как вызвать параметризованную функцию без параметров?</p> <p>Все ли компонентные функции параметризованного класса являются параметризованными?</p> <p>Являются ли дружественные функции, описанные в параметризованном классе, параметризованными?</p> <p>Могут ли шаблоны классов содержать виртуальные компонентные функции?</p>
3.	Задания к лабораторным работам	<p>Первый семестр</p> <p>Лабораторная работа №1</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>Арифметические операции. Условный оператор</p> <p>Разработать программу для вычисления заданных выражений и вывода полученных результатов. Соответствующие исходные данные ввести с клавиатуры.</p> <p>Вариант 1</p> $a = \ln(y^{-\sqrt{ x }}) \cdot (\sin(x) + e^{(x+y)})$ $b = \begin{cases} \ln(x/y) + (x^2 + y)^3, & x/y > 0 \\ \ln x/y + (x^2 + y)^3, & x/y < 0 \\ (x^2 + y)^3, & y \neq 0, x = 0 \\ 0, & y = 0 \end{cases}$ <p>Лабораторная работа №2</p> <p>Операторы цикла. Табулирование функции</p> <p>Вычислить и вывести на экран таблицу функции в интервале [a, b] с шагом h.</p> <p>Вариант 14</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		$y = \begin{cases} \frac{x^3}{3+x}, & x < -3 \\ 0, & x = 3 \\ 9x - \frac{27}{7} \ln(3+x), & \text{иначе} \end{cases}$ $x \in [-4, -2], h = 0.1$

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>заменить:</p> <p>для нечетных (по порядку) символов 3-й бит единицей; для четных символов – 4-й бит нулем.</p> <p>Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.</p> <p>Лабораторная работа №4</p> <p>Рекуррентные последовательности. Операторы цикла</p> <p>4.7. Вычислить произведение</p> $P = \prod_{i=1}^{15} U_i ,$ <p>где последовательность $\{U_i\}$ задается так:</p> $U_1 = 0.4;$ $U_2 = 0.5;$ $U_3 = 0.93;$ $U_i = \sin(U_{i-1}) + \cos(U_{i-3})$ <p>$i = 4, 5, \dots, 15.$</p> <p>Массивом не пользоваться.</p> <p>Лабораторная работа №5</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>Одномерные массивы</p> <p>5.1. Даны действительные числа a_1, \dots, a_{15}.</p> <p>Получить</p> $t = \frac{1}{15} \sum_{i=1}^{15} a_i, \quad s = \sqrt{\frac{\sum_{i=1}^{15} (a_i - t)^2}{14}}.$ <p>Лабораторная работа №6</p> <p>Двумерные массивы</p> <p>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.</p> <p>Лабораторная работа №7</p> <p>Работа с динамической памятью (операции <i>new</i> и <i>delete</i>)</p> <p>7.1. Получить квадратную матрицу порядка n:</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		$\begin{bmatrix} n & 0 & \dots\dots\dots & 0 \\ n-1 & n & 0 & \dots\dots & 0 \\ n-2 & n-1 & n & 0 & \dots & 0 \\ & & \dots & & & \\ 1 & 2 & 3 & \dots\dots & n \end{bmatrix}$ <p>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.</p> <p>Вывести полученную матрицу.</p> <p>Лабораторная работа №8 Работа с файлами, символьными массивами и строками</p> <p>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.</p> <p>Лабораторная работа №9 Функции. Передача аргументов. Аргументы по умолчанию</p> <p>Задание: Вычислить интеграл (для нечетных вариантов – методом средних прямоугольников, для четных – методом Симпсона).</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>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).</p> <p>Интеграл вычислить:</p> <ol style="list-style-type: none"> 1. С умалчиваемым значением параметра; 2. С другим значением этого параметра; 3. По формуле Ньютона-Лейбница. <p>Сравнить полученные значения.</p> $9.1. \int_1^{3.5} \frac{\ln x}{x\sqrt{1+\ln x}} dx$ <p>Лабораторная работа №10</p> <p>Системы счисления</p> <ol style="list-style-type: none"> 1. Дано натуральное число n. Определить, сколько различных цифр встречается в его десятичной записи. <p>Второй семестр</p> <p>Лабораторная работа №1</p> <p>Файлы и структуры</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>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 место жительства 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.</p> <p>Лабораторная работа №2</p> <p>Классы. Конструкторы и деструкторы</p> <p>1. Необходимо реализовать класс с двумя полями, имеющими имена first и second. Обязательно должны присутствовать:</p> <ul style="list-style-type: none"> • Для инициализации начальных значений использовать конструктор; • Обязательно использовать деструктор для удаления объектов; • Вывод на экран Display(); • Для проверки программы использовать несколько случайных наборов данных для переменных first и second. Вывод результата на экран консоли. <p>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.</p> <p>Лабораторная работа №3</p> <p>Наследование</p> <p>1. Создать базовый класс Car (машина), характеризуемый торговой маркой (строка), числом цилиндров, мощностью. Определить функции переназначения марки и изменения мощности.</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>Создать производный класс Lorry (грузовик), характеризующийся также грузоподъемностью кузова. Определить функцию изменения грузоподъемности.</p> <p>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.</p> <p>Лабораторная работа №4</p> <p>Переопределение стандартных операций</p> <p>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:</p> <p>сложения +, $(a, b) + (c, d) = (a + b, c + d)$;</p> <p>вычитания -, $(a, b) - (c, d) = (a - b, c - d)$;</p> <p>умножения *, $(a, b) * (c, d) = (ac - bd, ad + bc)$;</p> <p>деления / , $(a, b) / (c, d) = (ac + bd, bc - ad) / (c^2 + d^2)$;</p> <p>сравнение ==, $(a, b) = (c, d)$, если $(a = c)$ и $(b = d)$;</p> <p>Для демонстрации работы программы необходимо создать объект класса, задать все поля данного объекта и продемонстрировать использование всех доступных функций класса, например, в консоли.</p> <p>Лабораторная работа №5</p> <p>Шаблоны классов</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>1. Создать шаблон класса для работы с массивом произвольного типа данных. Шаблон должен включать:</p> <ul style="list-style-type: none"> • 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 <p>и другие необходимые для работы методы (например, переопределение операций индексирования и присваивания, методы ввода / вывода и т.д.).</p> <p>Задание:</p> <p>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.</p> <p>Лабораторная работа №6</p> <p>Динамические структуры данных. Списки, стеки, двоичные деревья</p> <p>2. Для решения задачи необходимо подобрать подходящую для этой задачи динамическую структуру данных.</p> <p>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##НЕЛО#ЛО должен быть напечатан в виде HELLO).</p>

	Оценочные мероприятия	Примеры типовых контрольных заданий
4.	Экзамен	<p>Вопросы тестирования (примеры):</p> <p>21. Выберите правильный набор символов, входящий в алфавит языка C++</p> <p>1) A { [] () b + - = ^ * . ф , / % \ ; : < = > ~ & _ "</p> <p>2) " { } , [] + () = ? ^ * . / \ ; ' : < = > ~ # & ! _ "</p> <p>3) { } [] () + - = ? ^ * . \$, / % \ ; : < = > ~ # & _ "</p> <p>4) [] () ? + - " { } = ? . , * ^ / % \ ; : @ < = > ~ # !</p> <p>22. Определите значение переменных a, b, c после выполнения следующих операций:</p> <pre>int a=0, b=1, c; c=a++; c=++a; c=++b;</pre> <p>В ответ запишите 3 числа через пробел (a b c)</p> <p>23. Определите значения переменных d1, d2, d3 после выполнения действий:</p> <pre>double d1,d2,d3; d1 = 2.56; d2 = (int)d1 + 1.5; d3 = (int)(d1 + 1.5);</pre> <p>24. Запишите значение переменной u после выполнения действий:</p> <pre>unsigned u2, u; int u1 = -1; u2 = -1; u=0; if (u1 != u2) u++;</pre> <p>25. void main()</p> <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> <p>При выполнении этого фрагмента кода на консоли появятся числа _____ <i>Запишите два числа через пробел</i></p> <p>26. int *a; int b; a = &b; b = 7; *a++; cout << b; system ("pause");</p> <p>После выполнения этого фрагмента программы на консоли появится число _____</p> <p>27. char x[] = "Hello world"; char* p = x; p += 4; cout << *p; system ("pause");</p> <p>После выполнения этого фрагмента программы на консоли появится</p> <ol style="list-style-type: none"> 1) o 2) o world 3) Hello world 4) ;5\$%%#0 <p>28. Прототип функции задает:</p> <ol style="list-style-type: none"> 5) тип функции, включая количество и тип аргументов и тип результата 6) возможность выполнения этой функции из программ на других языках программирования 7) имя функции и минимальное количество ее параметров 8) стандартная функция форматированного вывода, с помощью которой осуществляется вывод данных в стандартный поток <p>29. Укажите верное утверждение</p> <ol style="list-style-type: none"> 5) Две функции с одним и тем же числом, порядком и типом аргументов могут быть перегружены, если обе не имеют аргументов по умолчанию 6) Перегруженные функции не должны иметь параметры по умолчанию

	Оценочные мероприятия	Примеры типовых контрольных заданий
		<p>7) Перегруженные функции должны иметь параметры по умолчанию, начинающиеся слева в списке аргументов</p> <p>8) Функция может быть перегружена более, чем один раз</p> <p>30. Отметьте, какому определению функции может соответствовать вызов func(5.98)</p> <p>6) <code>int func(double x = 0, double y);</code> 7) <code>void func(double x);</code> 8) <code>double func(double x, int y = 12, int z = 5);</code> 9) <code>void func(float arg1, float arg2 = 0);</code> 10) <code>float func(float g, float f);</code></p> <p>31. Укажите, какая из записей соответствует обращению к члену m_arg класса AC в определении метода этого же класса</p> <p>5) <code>this.m_arg.AC</code> 6) <code>m_arg</code> 7) <code>AC.this.m_arg</code> 8) <code>AC->m_arg</code></p> <p>32. Абстрактный класс – это класс, в котором есть</p> <p>5) виртуальный конструктор 6) виртуальный деструктор 7) хотя бы одна чисто виртуальная функция 8) абстрактный класс определяет неявно некоторое множество объектов, задавая применимые к ним функции</p>

5. Методические указания по процедуре оценивания

	Оценочные мероприятия	Процедура проведения оценочного мероприятия и необходимые методические указания
1.	Экзамен	Экзамен проводится Центром обеспечения качества обучения ТПУ в соответствии с утвержденной программой и спецификацией в виде тестирования (независимого)
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 дисциплины