

**ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ  
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО  
ОБРАЗОВАНИЯ "ЦЕНТР ОБУЧЕНИЯ "СПЕЦИАЛИСТ" УНЦ ПРИ  
МГТУ ИМ. Н.Э. БАУМАНА  
(ОЧУ ДПО «СПЕЦИАЛИСТ»)**

123317, г. Москва, Пресненская набережная, д 8, стр. 1, этаж 48, помещение 484с, комната 3,  
ИНН 7701168244, ОГРН 1127799002990

---

Утверждаю:

Директор ОЧУ ДПО «Специалист»



/Е.В. Добрыднева/  
февраля \_\_\_\_ 2018 года

**Дополнительная профессиональная программа  
повышения квалификации  
Программирование на языке С (Си)**

город Москва

Программа разработана в соответствии с приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. N 499 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам".

Повышение квалификации слушателей, осуществляемое в соответствии с программой, проводится с использованием модульного принципа построения учебного плана с применением различных образовательных технологий, в том числе дистанционных образовательных технологий и электронного обучения в соответствии с законодательством об образовании.

Дополнительная профессиональная программа повышения квалификации, разработана образовательной организацией в соответствии с законодательством Российской Федерации, включает все модули, указанные в учебном плане.

Содержание оценочных и методических материалов определяется образовательной организацией самостоятельно с учетом положений законодательства об образовании Российской Федерации.

Структура дополнительной профессиональной программы соответствует требованиям Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденного приказом Минобрнауки России от 1 июля 2013 г. N 499.

Объем дополнительной профессиональной программы вне зависимости от применяемых образовательных технологий, должен быть не менее 16 академических часов. Сроки ее освоения определяются образовательной организацией самостоятельно.

Формы обучения слушателей (очная, очно-заочная, заочная) определяются образовательной организацией самостоятельно.

К освоению дополнительных профессиональных программ допускаются:

- лица, имеющие среднее профессиональное и (или) высшее образование;
- лица, получающие среднее профессиональное и (или) высшее образование.

Для определения структуры дополнительной профессиональной программы и трудоемкости ее освоения может применяться система зачетных единиц. Количество зачетных единиц по дополнительной профессиональной программе устанавливается организацией.

Образовательная деятельность слушателей предусматривает следующие виды учебных занятий и учебных работ: лекции, практические и семинарские занятия, лабораторные работы, круглые столы, мастер-классы, мастерские, деловые игры, ролевые игры, тренинги, семинары по обмену опытом, выездные занятия, консультации, выполнение аттестационной, дипломной, проектной работы и другие виды учебных занятий и учебных работ, определенные учебным планом.

### 1. Цель программы:

В результате прохождения обучения слушатель научится использовать все операторы языка C, препроцессорные директивы, пользоваться библиотечными функциями. Освоите работу с массивами, структурами данных и с файлами.

#### Совершенствуемые компетенции

№	Компетенция	Направление подготовки
		ФГОС ВО ПО НАПРАВЛЕНИЮ ПОДГОТОВКИ 09.03.02 «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ» (УРОВЕНЬ

		БАКАЛАВРИАТА)
		Код компетенции
1	способностью использовать технологии разработки объектов профессиональной деятельности в областях: машиностроение, приборостроение, техника, образование, медицина, административное управление, юриспруденция, бизнес, предпринимательство, коммерция, менеджмент, банковские системы, безопасность информационных систем, управление технологическими процессами, механика, техническая физика, энергетика, ядерная энергетика, силовая электроника, металлургия, строительство, транспорт, железнодорожный транспорт, связь, телекоммуникации, управление инфокоммуникациями, почтовая связь, химическая промышленность, сельское хозяйство, текстильная и легкая промышленность, пищевая промышленность, медицинские и биотехнологии, горное дело, обеспечение безопасности подземных предприятий и производств, геология, нефтегазовая отрасль, геодезия и картография, геоинформационные системы, лесной комплекс, химико-лесной комплекс, экология, сфера сервиса, системы массовой информации, дизайн, медиаиндустрия, а также предприятия различного профиля и все виды деятельности в условиях экономики информационного общества (ПК-17);	ПК-17
2	способностью поддерживать работоспособность информационных систем и технологий в заданных функциональных характеристиках и соответствии критериям качества (ПК-30);	ПК-30
3	способностью обеспечивать безопасность и целостность данных информационных систем и технологий	ПК-31

**Совершенствуемые компетенции в соответствии с трудовыми функциями профессионального стандарта «ПРОГРАММИСТ», утвержденного приказом Минтруда и социальной защиты РФ от 18 ноября 2013 г. N 679н**

№	Компетенция	Направление подготовки
		Трудовые функции (код)
1	Разработка и отладка программного кода (Формализация и алгоритмизация поставленных задач, Написание программного кода с использованием языков программирования, определения и манипулирования данными, Оформление программного кода в соответствии с установленными требованиями)	А/01.3; А/02.3; А/03.3

**Планируемый результат обучения:**

**После окончания обучения Слушатель будет знать:**

1. Лексемы и пробельные символы.
2. Система типов языка С.
3. Диапазоны представляемых значений.
4. Декларация переменных.

5. Литералы.
6. Использование библиотечных функций.
7. Написание простых программ.
8. Компиляция, компоновка и выполнение программ.

**После окончания обучения Слушатель будет уметь:**

9. использовать все операторы языка C;
10. использовать препроцессорные директивы;
11. пользоваться библиотечными функциями;
12. работать с массивами;
13. писать функции;
14. использовать классы памяти;
15. применять адресную арифметику;
16. работать со структурами данных;
17. работать с файлами.

**Учебный план:**

Категория слушателей: курс предназначен для:

- Руководители проектов и отделов;
- Руководители линейных подразделений;
- Помощники руководителя проектов
- Программисты
- Разработчики и аналитики компьютерных систем

Требования к предварительной подготовке:

Успешное окончание курса «Основы программирования и баз данных»

Срок обучения: 48 академических часов, 24 самостоятельно

Форма обучения: очная, очно-заочная, заочная. По желанию слушателя форма обучения может быть изменена и/или дополнена.

Режим занятий: дневной, вечерний, группы выходного дня.

№ п/п	Наименование модулей по программе	Общая трудоемкость (акад. часов)	Всего ауд.ч	В том числе		СРС,ч
				Лекций	Практических занятий	
1	Модуль 1. Введение в язык Си	6	4	2	2	2
2	Модуль 2. Структура программы	6	4	2	2	2
3	Модуль 3. Операции	6	4	1	3	2
4	Модуль 4. Операторы	6	4	2	2	2
5	Модуль 5. Препроцессор	6	4	2	2	2

6	Модуль 6. Массивы	6	4	2	2	2
7	Модуль 7. Функции	6	4	2	2	2
8	Модуль 8. Классы памяти	6	4	2	2	2
9	Модуль 9. Адресная арифметика	6	4	2	2	2
10	Модуль 10. Файлы	6	4	2	2	2
11	Модуль 11. Структуры	6	4	2	2	2
12	Модуль 12. Итоговое занятие	6	4	2	2	2
	Итого:	72	48	24	24	24
	Итоговая аттестация	тестирование				

Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут.

## 2. Календарный учебный график

Календарный учебный график формируется при осуществлении обучения в течение всего календарного года. По мере набора групп слушателей по программе составляется календарный график, учитывающий объемы лекций, практики, самоподготовки, выезды на объекты.

Неделя обучения	1	2	3	4	5	6	7	Итого часов
	пн	вт	ср	чт	пт	сб	вс	
1 неделя	8	8	8	8	8	-	-	40
2 неделя	8ИА	-	-	-	-	-	-	8
СРС	8	4	4	4	4			24
Итого:								72

Примечание: ИА – Итоговая аттестация (лабораторная работа, контрольные вопросы)

### Рабочие программы учебных предметов

#### Модуль 1. Введение в язык Си

- Лексемы и пробельные символы.
- Система типов языка С.
- Диапазоны представляемых значений.
- Декларация переменных.
- Литералы.

#### Модуль 2. Структура программы

- Использование библиотечных функций.
- Ввод с клавиатуры и вывод на экран.
- Написание простых программ.
- Компиляция, компоновка и выполнение программ.

### **Модуль 3. Операции.**

- Арифметические и логические операции.
- Операции присваивания.
- Отношения.
- Битовые, адресные и прочие операции.
- Приоритеты и ассоциативность операций.
- Трассировка программы.

### **Модуль 4. Операторы**

- Оператор-выражение и составной оператор.
- Метки.
- Управляющие операторы – goto, if-else, switch.
- Операторы цикла – while, do-while и for.
- Использование операторов break, continue, и return.

### **Модуль 5. Препроцессор**

- Препроцессорные директивы #include, #define, #undef, #if - #else - #endif.
- Макроопределения с параметрами.
- Правила оформления деклараций.

### **Модуль 6. Массивы**

- Декларация массивов и их размещение в памяти.
- Индексация элементов массива.
- Массивы переменной длины (VLA).
- Инициализация массивов.
- Алгоритмы суммирования, поиска и сортировки.

### **Модуль 7. Функции**

- Аргументы и параметры.
- Прототип функции.
- Возвращение значения функцией.
- Рекурсия.
- Выражения, не зависящие от типа (\_Generic).

### **Модуль 8. Классы памяти**

- Время жизни и область видимости объекта.
- Декларации на внутреннем и внешнем уровнях.
- Модификаторы – auto, register, static, extern.
- Динамическое распределение памяти.
- Определяемые типы typedef.

### **Модуль 9. Адресная арифметика**

- Декларация указателей.
- Операции с указателями.
- Использование указателей как аргументов функции.
- Указатели на функции.
- Указатели и массивы
- Указатели и строки.

## **Модуль 10. Файлы**

- Понятие потока – stream.
- Структура типа FILE.
- Открытие и закрытие файлов.
- Операции ввода-вывода.
- Указатель чтения-записи в файле.
- Последовательный и произвольный доступ.

## **Модуль 11. Структуры**

- Декларация структур.
- Инициализация и доступ к элементам структуры.
- Вложенные структуры и массивы структур.
- Выравнивание данных `_Alignas`, `_Alignof`
- Объединения.

## **Модуль 12. Итоговое занятие**

- Выполнение курсовой работы.

## **Организационно-педагогические условия**

Соблюдение требований к кадровым условиям реализации дополнительной профессиональной программы:

а) преподавательский состав образовательной организации, обеспечивающий образовательный процесс, обладает высшим образованием и стажем преподавания по изучаемой тематике не менее 1 года и (или) практической работы в областях знаний, предусмотренных модулями программы, не менее 3 (трех) лет;

б) образовательной организацией наряду с традиционными лекционно-семинарскими занятиями применяются современные эффективные методики преподавания с применением интерактивных форм обучения, аудиовизуальных средств, информационно-телекоммуникационных ресурсов и наглядных учебных пособий.

Соблюдение требований к материально-техническому и учебно-методическому обеспечению дополнительной профессиональной программы:

а) образовательная организация располагает необходимой материально-технической базой, включая современные аудитории, библиотеку, аудиовизуальные средства обучения, мультимедийную аппаратуру, оргтехнику, копировальные аппараты. Материальная база соответствует санитарным и техническим нормам и правилам и обеспечивает проведение всех видов практической и дисциплинарной подготовки слушателей, предусмотренных учебным планом реализуемой дополнительной профессиональной программы.

б) в случае применения электронного обучения, дистанционных образовательных технологий каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронной информационно-

образовательной среде, содержащей все электронные образовательные ресурсы, перечисленные в модулях дополнительной профессиональной программы.

### **3. Формы аттестации и оценочные материалы**

Образовательная организация несет ответственность за качество подготовки слушателей и реализацию дополнительной профессиональной программы в полном объеме в соответствии с учебным планом.

Оценка качества освоения дополнительной профессиональной программы слушателей включает текущий контроль успеваемости и итоговую аттестацию.

Конкретные формы и процедуры текущего контроля успеваемости, промежуточной аттестации и итоговой аттестации слушателей устанавливаются образовательной организацией самостоятельно.

Слушателям, успешно освоившим дополнительную профессиональную программу и прошедшим итоговую аттестацию, выдается удостоверение о повышении квалификации.

Слушателям, не прошедшим итоговой аттестации или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть дополнительной профессиональной программы и (или) отчисленным из образовательной организации, выдается справка об обучении или о периоде обучения по образцу, самостоятельно устанавливаемому образовательной организацией.

Результаты итоговой аттестации слушателей ДПП в соответствии с формой итоговой аттестации, установленной учебным планом, выставляются по двух бальной шкале («зачтено\незачтено»).

Итоговая аттестация проводится по форме выполнения лабораторной работы и аттестации по контрольным вопросам в соответствии с учебным планом. Результаты итоговой аттестации заносятся в соответствующие документы.

## **Итоговая аттестация**

### **Лабораторная работа. Структура программы, основные типы данных, ввод/вывод**

**Цель работы** – изучить структуру программы, научиться использовать переменные различных типов, освоить функции форматного ввода и вывода, арифметические операции и операции присваивания.

#### **Постановка задачи**

1. Набрать текст программы, представленный ниже. Проанализировать значения переменных после каждой операции присваивания. Проверить порядок выполнения операций в каждом выражении, содержащем несколько операций присваивания, разделив каждый оператор-выражение на несколько операторов, выполняемых последовательно. В функциях ввода и вывода изменить спецификаторы формата, проанализировать полученные результаты.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main (void)
```

```
{
```



```

int a, b = 5, c;

double x, y = -.5, z;

printf("a=");

scanf("%d", &a);

x = c = a;

printf("x = c = a : a=%d c=%d x=%f\n", a, c, x);

a += b;

printf("a += b : a=%d\n", a);

x *= b+a;

printf("x *= b+a : x=%lf\n", x);

b += a--;

printf("b += a-- : a=%d b=%d\n", a, b);

x -= ++c;

printf("x -= ++c : c=%d x=%lf\n", c, x);

c = a/b;

printf("c = a/b : c=%4d\n", c);

c = a%b;

printf("c = a%%b : c=%d\n", c);

y += (a+1)/a++;

printf("y += (a+1)/a++ : a=%d y=%.3lfty=%.0lf\n", a, y, y);

b = 3*(y-=.6)+2*b+1;

printf("b = 3*(y-=.6)+2*b+1 : b=%d y=%.1lf\n", b, y);

z = a/2;

printf("z = a/2 : z = a/2 : z=%lf\n", z);

z = (double)a/2;

printf("z = (double)a/2 : z=%lf\n", z);

y = (x = 5.7)/2;

printf("y = (x = 5.7)/2 : x=%lf y=%lf\n", x, y);

```

```

y = (int)x/2;
printf("y = (int)x/2 : y=%f\n", y);
z = (b-3)/2 - x/5 +(c/=2) + 1/4*z - y++ + ++b/3.;
printf("z = (b-3)/2 - x/5 +(c/=2) + 1/4*z - y++ + ++b/3. :\n\
a=%d b=%d c=%d x=%lf y=%lf z=%lf\n", a,b,c,x,y,z);
system("pause");
return 0;
}

```

2. Написать программу для вычисления значений следующих выражений:  $a=5, c=5$

$a=a+b-2$

$c=c+1, d=c-a+d$

$a=a*c, c=c-1$

$a=a/10, c=c/2, b=b-1, d=d*(c+b+a)$

Выражения, записанные в одной строке, записывать одним оператором-выражением. Переменные  $a, b, c, d$  объявить как целые, переменные  $e, a, i, b$  – как вещественные. Значения переменных  $b, i, d$  вводить с клавиатуры. После вычисления каждого выражения выводить на экран значения всех переменных.

3. Набрать текст программы, представленный ниже. Проанализировать выдаваемые программой результаты. Объяснить, почему они именно такие.

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <limits.h>
```

```
#include <float.h>
```

```
main()
```

```
{
```

```
char c;
```

```
unsigned char uc;
```

```
int i;
```

```
unsigned u;
```

```
short s;
```

```

long l;

float f;

double d;

printf("sizeof(c)=%d\tsizeof(uc)=%d\nsizeof(i)=%d\tsizeof(u)=%d\t\
sizeof(s)=%d\tsizeof(l)=%d\nsizeof(f)=%d\tsizeof(d)=%d\n\n",
sizeof(c),sizeof(uc),sizeof(i),sizeof(u),sizeof(s),
sizeof(l),sizeof(f),sizeof(d));

uc=c=CHAR_MAX;

printf("CHAR_MAX : c=%d uc=%d\n", c, uc);

c++; uc++;

printf("CHAR_MAX+1 : c=%d uc=%d\n", c, uc);

uc=c=CHAR_MIN;

printf("CHAR_MIN : c=%d uc=%d\n", c, uc);

c=uc=UCHAR_MAX;

printf("UCHAR_MAX : c=%d uc=%d\n", c, uc);

c++; uc++;

printf("UCHAR_MAX+1 : c=%d uc=%d\n", c, uc);

uc=c=-5;

printf("-5 : c=%d uc=%d\n", c, uc);

c=-5; uc=5;

printf("char and unsigned char -5>5 : %d\n\n", c>uc);

c=s=SHRT_MAX;

uc=s;

printf("SHRT_MAX : c=%d uc=%d s=%d\n", c, uc, s);

s++;

printf("SHRT_MAX+1 : s=%d\n", s);

c=s; uc=s;

printf("%d : c=%d uc=%d\n", SHRT_MIN, c, uc);

```

```
s=0; c=s; uc=s;
printf("0 : c=%d uc=%d s=%d\n", c, uc, s);
i=INT_MAX;
l=i; u=i;
printf("INT_MAX : i=%d u=%u l=%ld\n", i, u, l);
i++; l++; u++;
printf("INT_MAX+1 : i=%d u=%u l=%ld\n", i, u, l);
i=INT_MIN;
l=i; u=i;
printf("INT_MIN : i=%d u=%u l=%ld\n", i, u, l);
u=UINT_MAX;
i=u; l=u;
printf("UINT_MAX : i=%d u=%u l=%ld\n", i, u, l);
u=i-5;
printf("-5 : i=%d u=%u\n", i, u);
i=-5; u=5;
printf("int and unsigned int -5>5 : %d\n", i>u);
c=-5; u=5;
printf("char and unsigned int -5>5 : %d\n\n", c>u);
i=5.1;
printf("i=5.1 : i=%d\n", i);
i=5.9;
printf("i=5.9 : i=%d\n", i);
d=f=FLT_MAX;
printf("FLT_MAX : f=%g d=%g\n", f, d);
d=f=FLT_MIN;
printf("FLT_MIN : f=%g d=%g\n", f, d);
d=f=FLT_EPSILON;
```

```
printf("FLT_EPSILON : f=%g d=%g\n", f, d);
```

```
f=1e10;
```

```
printf("1e10 : f=%f\n", f);
```

```
f=1e11;
```

```
printf("1e11 : f=%f\n", f);
```

```
f=1234567890;
```

```
printf("1234567890 : f=%f\n", f);
```

```
d=DBL_MAX;
```

```
printf("DBL_MAX : d=%g\n", d);
```

```
d=DBL_MIN;
```

```
printf("DBL_MIN : d=%g\n", d);
```

```
d=DBL_EPSILON;
```

```
printf("DBL_EPSILON : d=%g\n", d);
```

```
d=1e15+1;
```

```
printf("1e15+1 : d=%lf\n", d);
```

```
d=1e16+1;
```

```
printf("1e16+1 : d=%lf\n", d);
```

```
f=10000*100000;
```

```
f+=1;
```

```
f-=4*250000000;
```

```
printf("1 : f=%f\n", f);
```

```
f=10000*100000+1-4*250000000;
```

```
printf("1 : f=%f\n", f);
```

```
d=10000*100000+1-4*250000000;
```

```
printf("1 : d=%lf\n", d);
```

```
d=1e20*1e20+1000-1e22*1e18;
```

```
printf("1000 : d=%lf\n", d);
```

```
system("pause");
```

```
return 0;  
  
}
```

## Контрольные вопросы

1. Какова структура программы на языке Си?
2. Зачем нужна директива `#include`?
3. Что такое `main()`?
4. Перечислите скалярные типы данных языка Си.
5. Что определяет тип данного?
6. Что такое `void`?
7. Что такое явное и неявное приведение типов? Как и когда оно используется?
8. Что такое константа? Найдите константы в набранных вами программах.
9. Что такое переменная?
10. Как проинициализировать переменную?
11. Чем отличается оператор от операции?
12. Чем отличаются унарные операции от бинарных?
13. Какие операции относятся к арифметическим? Каков приоритет каждой из них?
14. Каков порядок выполнения операций в случае их одинакового приоритета?
15. Как выполняется операция деления в случае целочисленных операндов и в случае, когда хотя бы один из операндов вещественный?
16. Что такое выражение?
17. Какое значение вычисляет операция присваивания?
18. В каком порядке выполняются присваивания в случае, если в выражении их несколько?
19. Как и зачем используются дополнительные операции присваивания?
20. Чем отличается префиксная форма операции инкремента или декремента от постфиксной?
21. Какие функции используются для ввода информации? Назовите их отличительные особенности.
22. Какие функции используются для вывода информации? Назовите их отличительные особенности.
23. Почему функции `scanf()` и `printf()` называются функциями форматного ввода и вывода? Как они работают?
24. Чем отличается управляющая строка функции `scanf()` от управляющей строки функции `printf()`?
25. Что такое спецификатор формата? Зачем он нужен?
26. Какие параметры указываются функции `scanf()` после управляющей строки? Сколько их должно быть?
27. Каковы последствия несоответствия типа считываемой функцией `scanf()` переменной спецификатору типа?
28. Какие параметры указываются функции `printf()` после управляющей строки? Сколько их должно быть?
29. Каковы последствия несоответствия типа выводимого функцией `printf()` значения спецификатору типа?
30. Что такое управляющие символы? Зачем они нужны? Приведите примеры.