



# **ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C++**

```
/* Пример простой программы*/  
#include  
int main()  
{  
printf(“Hello World!”);  
return 0;  
}
```


# 1.1. Структура и этапы создания программы на языке C++

В самом простом случае этот текстовый файл может содержать такую информацию:


Листинг 1.1. Пример простой программы.

```
/* Пример простой программы*/  
#include  
int main()  
{  
printf(“Hello World!”);  
return 0;  
}
```

и обычно имеет расширение `cpp`, например, «`ex1.cpp`».



Компилятор создает не готовую к исполнению программу, а только объектный код (файл с расширением \*.obj).






В приведенной программе используется функция `printf()` стандартной библиотеки «`stdio.h`».

Это означает, что объектный файл `ex1.obj` будет содержать лишь инструкции по вызову данной функции, но код самой функции в нем будет отсутствовать.



## 1.2. Стандарты языка C++

- Американский институт национальных стандартов (American National Standards Institute (ANSI)) в 1983 г. организовал комитет для разработки нового стандарта, который был принят в 1989 г.
- Этот стандарт (ANSI C) содержит определение как языка, так и стандартной библиотеки C.
- Затем международная организация по стандартизации (ISO) в 1990 г. приняла свой стандарт (ISO C), который по сути не отличается от стандарта ANSI C.

- 
- В 1994 г. возобновилась деятельность по разработке нового стандарта, в результате чего появился стандарт C99, который соответствует языку C++.
  - Объединенный комитет ANSI/ISO развил исходные принципы предыдущего стандарта, являющийся основным на сегодняшний день.

# 1.3. Представление данных в языке C++

<тип переменной> <имя\_переменной>;

Например, строка

```
int arg;
```

## Основные базовые типы данных языка C++:

- int - целочисленный тип 16 либо 32 бит
- long int - целочисленный тип 32 бит
- short - целочисленный тип 8 либо 16 бит
- char - символьный тип 8 бит
- float - вещественный тип 32 бит
- double - вещественный тип 64 бит



## 1.4. Оператор присваивания

```
int length = 5;
```

---

```
int i=2;  
i=i+1;
```

---

```
20=i;
```





## Пример 1:

```
short agr_short = -10;  
long arg_long;
```

и выполняется оператор присваивания:

```
arg_long = arg_short;
```

## Пример 2:

```
float agr_f = 8.7;  
int arg_long;  
arg_long = agr_f;
```




## Синтаксис операции приведения ТИПОВ

<имя\_переменной\_1> = (тип\_данных)<имя\_переменной\_2>;

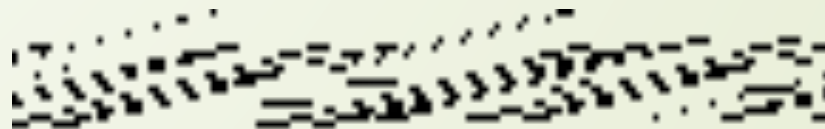
Например:

```
arg_long = (long )arg_f;
```

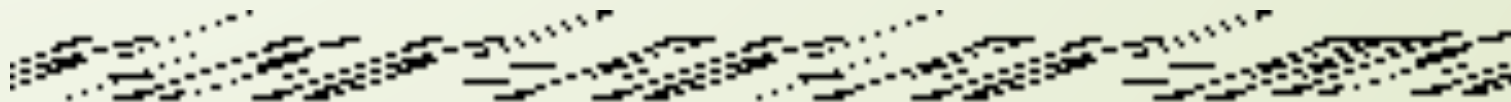


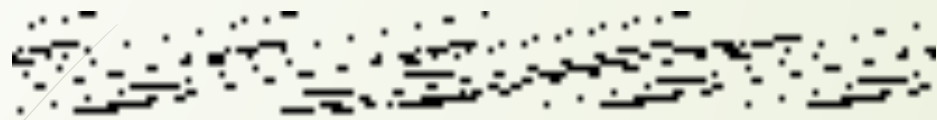


Для представления числа 16 используется  
шестнадцатиричное



а число 255 соответствует числу:





Например,

FF = 11111111; 0F = 00001111; 11 = 00010001 и т.д.

Представление шестнадцатиричных чисел в языке C++

```
int var = 0xff; //число 255
```

```
int arg = 0xac; //число 172
```

## 1.6. Арифметические операции

```
int c;  
c = a+b; //сложение двух переменных  
c = a-b; //вычитание  
c = a*b; //умножение  
c = a/b; //деление
```

---

```
c = 10+5;  
c = 8*4;  
float d;  
d = 7/2;
```

## корректное деление:

`d = 7.0/2;`

или

`d = (float )7/2;`

---

`int a,b;`

`a = 7; b = 2;`

`d = a/b;`

В результате значение d будет равно 3

---

`d = (float )a/b;`

d будет равно 3,5



Приоритет арифметических операций:

```
double n=2, SCALE = 1.2;  
double arg = 25.0 + 60.0*n/SCALE;
```

Поменяем приоритеты:

```
double arg = (25.0 + 60.0)*n/SCALE;
```





# Операция деления по модулю:

```
int a = 13 % 5;
```

Эта же операция на основе стандартных арифметических операций:

```
int a = 13 - 13/5*5;
```



# Компактные операторы инкремента и декремента

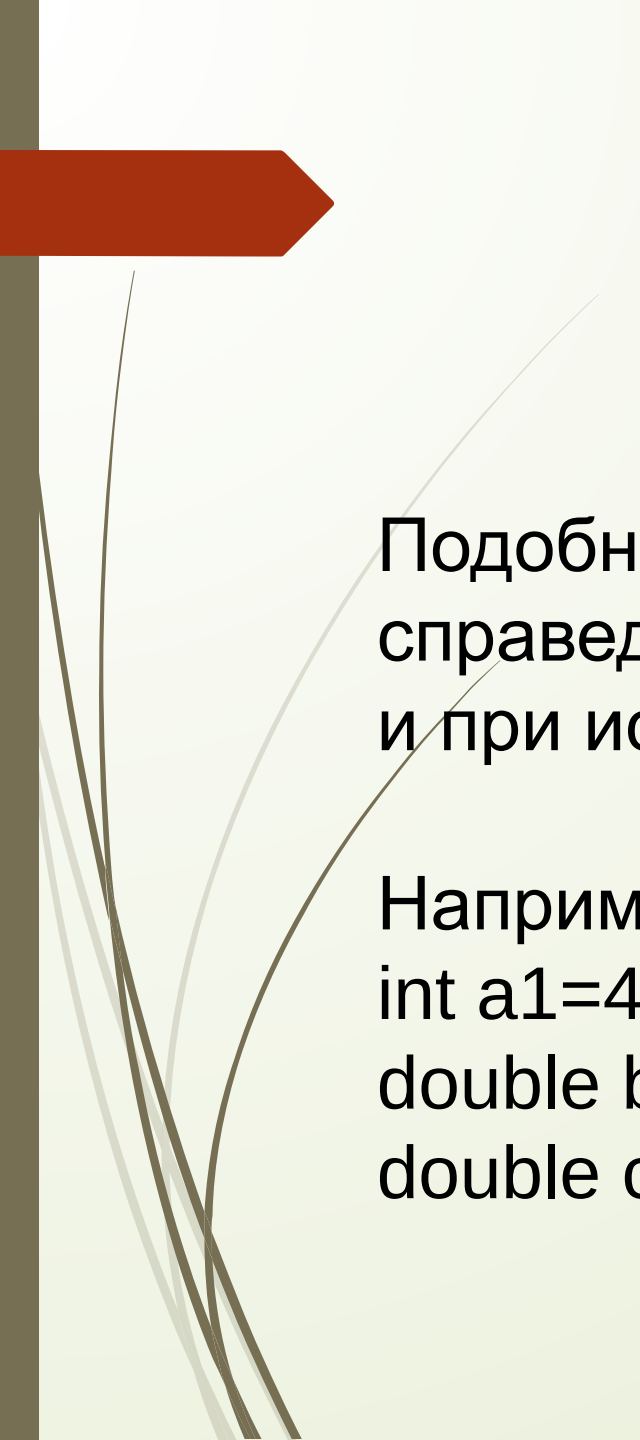
`i++;` // операция инкремента  
`++i;` // операция инкремента  
`i--;` // операция декремента  
`--i;` // операция декремента

Разница между первой и второй формами записи данных операторов продемонстрируем на примере:

```
int i=10,j=10;
```

```
int a = i++; //значение a = 10; i = 11;
```

```
int b = ++j; //значение b = 11; j = 11;
```



Подобный приоритет операции инкремента остается справедливым и при использовании арифметических операций

Например:

```
int a1=4, a2=4;
```

```
double b = 2.4*++a1; //результат b = 12.0
```

```
double c = 2.4*a2++; //результат c = 9.6
```



# 1.7. Поразрядные операции языка C++