Chapter 1: Starting with C++

Visual Studio

To learn more about privacy, see the Microsoft Privacy Statement.

By continuing, you agree to the License Terms.
Visual Studio

Products

Installed

Visual Studio Community 2017
Free, fully-featured IDE for students, open-source and individual developers

Available

Visual Studio Enterprise 2017
Microsoft DevOps solution for productivity and coordination across teams of any size

Visual Studio Professional 2017
Professional developer tools and services for small teams
Welcome!
Connect to all your developer services.

Sign in to start using your Azure credits, publish code to a private Git repository, sync your settings, and unlock the IDE.

Learn more

Not now, maybe later.
New File

Installed

- General
- Performance
- Web
  - Visual C++
- Script
- Graphics

Type: Visual C++
Creates a file containing C++ source code

- C++ File (.cpp) Visual C++
- Header File (.h) Visual C++
- C++ Properties for Open... Visual C++
Chapter 3: Exploring C++ Types

![Diagram showing C++ types and padding]

- uc
- us
- ui
- ull

**Padding**
C:\Beginning_C++\Chapter_03>hexdump hexdump.exe 10
4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00
b8 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68
69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f
74 20 62 65 20 72 75 6e 64 69 66 6f 6e 6f 6f
6f 20 66 6f 6f 6f 6e 6f 6f 6c 65 73 73 69 6e 67
2b c4 3f 01 6f a5 51 52 6f a5 51 52 6f a5 51 52
db 39 a0 52 62 a5 51 52 db 39 a2 52 fa a5 51 52

C:\Beginning_C++\Chapter_03>
Chapter 4: Working with Memory, Arrays, and Pointers

<table>
<thead>
<tr>
<th>Memory Address</th>
<th>Contents</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>007ef880</td>
<td>8c f8 7e 00</td>
<td>pi</td>
</tr>
<tr>
<td>007ef884</td>
<td></td>
<td></td>
</tr>
<tr>
<td>007ef888</td>
<td></td>
<td></td>
</tr>
<tr>
<td>007ef88c</td>
<td>2a 00 00 00</td>
<td>i</td>
</tr>
</tbody>
</table>

Before

<table>
<thead>
<tr>
<th></th>
<th>01</th>
<th>00</th>
<th>00</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>pv</td>
<td>02</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>05</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

After

<table>
<thead>
<tr>
<th></th>
<th>0b</th>
<th>00</th>
<th>00</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>pv</td>
<td>0c</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>v[1]</td>
<td>0d</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>pv[2]</td>
<td>0e</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>(pv + 3)</td>
<td>05</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>
Chapter 7: Introduction to Object-Orientated Programming

class base
{ public: int x = 0; };

class derived : public base
{ public: int y = 0; };

derived d;
```cpp
// base class
class base {public: int x;};

// derived from base
class base1 : public base{);
class base2 : public base{);

// derived class
class derived
  : public base1, public base2
{);

// derived object
{ }
```

```cpp
// base1
int x = 1;

// base2
int x = 2;

// derived object
{ int base1::x = 1;
  int base2::x = 2;
  // derived }
```

```cpp
// derived
{ int base1::x;
  int base2::x;
  // derived }
```
class base
{
public:
    virtual void fn1();
    virtual void fn2();
};
void base::fn1(){ }
void base::fn2(){ }

class derived : public base
{
public:
    fn2();
};
void derived::fn2(){ }

---

class base
{
public:
    virtual void fn1();
    virtual void fn2();
};

class derived : public base
{
public:
    virtual void fn1();
    // no fn2
    virtual void fn3();
};
```cpp
class base1
{
    public:
        virtual void fn1();
};

class base2
{
    public:
        virtual void fn2();
};

class derived1
    : public base1, public base2
{
    public:
        virtual void fn1();
        virtual void fn2();
};
```
Chapter 8: Using the Standard Library Containers

num1
2 7 1 8 2 8

num2
3 1 4 5 6 8

swap

num1
3 1 4 5 6 8

num2
2 7 1 8 2 8