ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп’ютерних

і телекомунікаційних систем

Кафедра інженерії програмного забезпечення

**Лабораторна робота №** 4

з дисципліни «Веб програмування»

на тему:

# «БЛОЧНА МОДЕЛЬ ДОКУМЕНТА»

**Виконав:**

студент 3 курсу, групи ІПЗ-17-1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ І. С. Мисик

(підпис) (Ініціали, прізвище)

**Перевірив:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Яшина О. Н.

(підпис)

Хмельницький – 2020 р.

Хід роботи

Варіант 12 – тема «C++».

Завдання:

Підготувати сторінку, що є енциклопедичною довідкою з теми. За зразок можна взяти структуру сторінки Вікіпедії.

При створенні сторінки використати отримані навички:

− структурувати сторінку, розділивши її на змістовні блоки (шапка, підвал, скорочена довідка в правій колонці, підрозділи);

− оформити текст, застосовуючи заголовки, списки, абзаци, зображення. Кожен елемент повинен мати стилі, всі стилі мають бути винесені в таблицю зв’язаних стилів;

− у підвал сторінки додати власне прізвище, що є посиланням на сторінку-резюме, розроблену в попередніх роботах.

Скопіювати проект та розділити сторінку, створену в попередньому пункті, на декілька сторінок, кожна з яких містить окремий підрозділ:

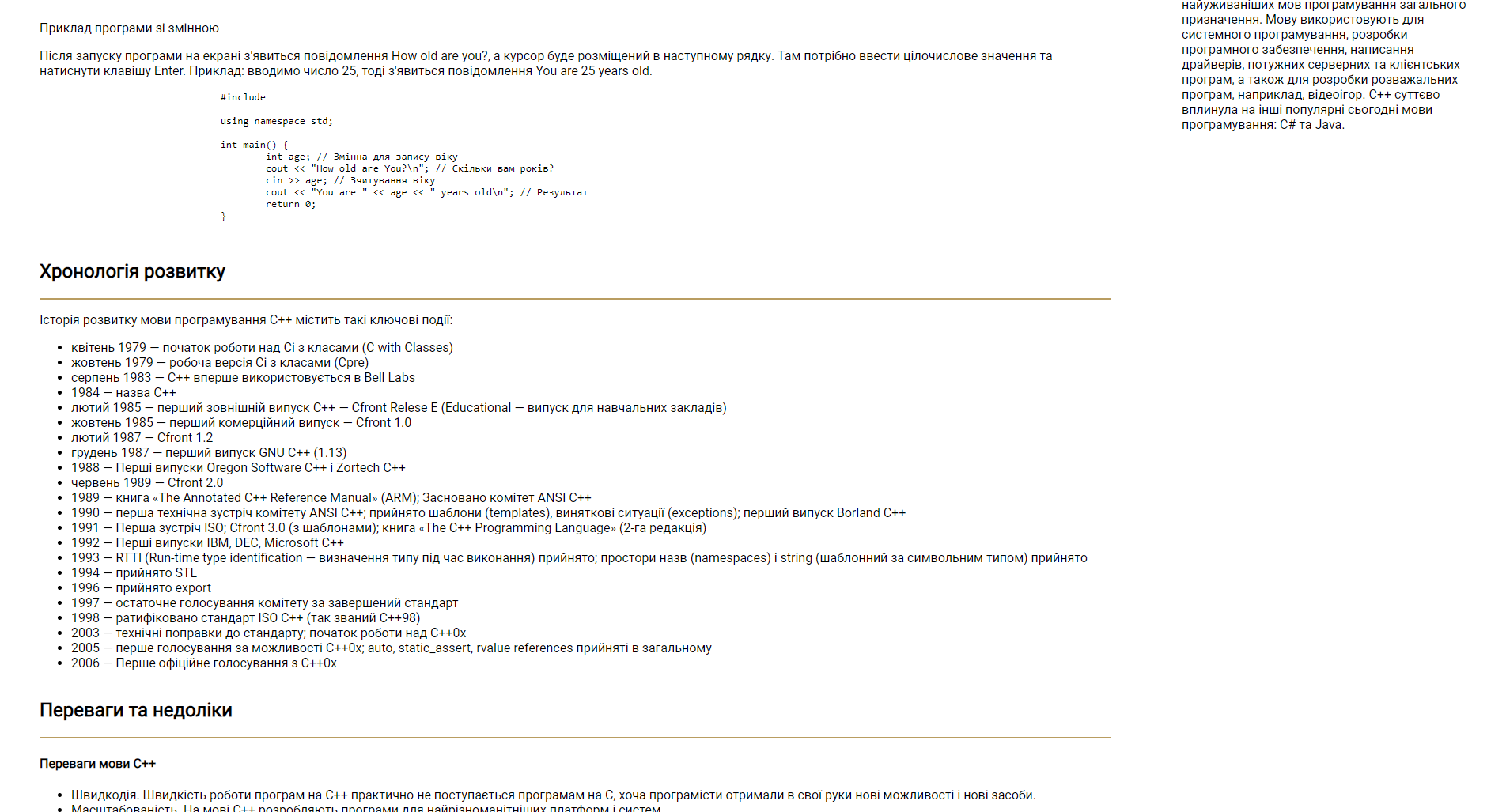
− на кожній зі сторінок зберегти бокову колонку зі скороченою довідкою;

− на всіх сторінках між шапкою та основним змістом додати горизонтальне навігаційне меню, що містить посилання на створені сторінки;

− виділяти стилем поточну сторінку в навігаційному меню.

Скріншот сторінки:







Код сторінки:

<!doctype html>  
<html lang="uk">  
<head>  
 <meta charset="UTF-8">  
 <title>С++</title>  
 <link rel="stylesheet" href="../css/block.css">  
 <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">  
</head>  
<body>  
 <header>  
 <h1>С++</h1>  
 </header>  
 <main id="main">  
 <p class="from">Матеріал з <a class="from\_1" href="https://uk.wikipedia.org/wiki/C%2B%2B">Вікіпедії</a> — вільної енциклопедії.</p>  
 <nav class="widget">  
 <h2 class="widget-title">Навігація</h2>  
 <ul class="widget-list">  
 <li><a href="block.html">Головна</a></li>  
 <li><a href="features.html">Особливості</a></li>  
 <li><a href="example.html">Приклад програми</a></li>  
 <li><a href="chronology.html">Хронологія розвитку</a></li>  
 <li><a href="advantages.html">Переваги та недоліки</a></li>  
 </ul>  
 </nav>  
 <!--<div id="mw-content-text" lang="uk" dir="ltr" class="mw-content-ltr"><div class="mw-parser-output"><table align="right" class="infobox" style="border-spacing: 1px; padding: 0px 0px; width: 280px; font-size:85%; line-height:130%;" cellspacing="1px">  
 <caption><big><b>C++</b></big>  
 </caption><tbody><tr><td colspan="2" style="text-align: center;"><a href="/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:ISO\_C%2B%2B\_Logo.svg" class="image"><img alt="ISO C++ Logo.svg" src="//upload.wikimedia.org/wikipedia/commons/thumb/1/18/ISO\_C%2B%2B\_Logo.svg/130px-ISO\_C%2B%2B\_Logo.svg.png" decoding="async" width="130" height="146" srcset="//upload.wikimedia.org/wikipedia/commons/thumb/1/18/ISO\_C%2B%2B\_Logo.svg/195px-ISO\_C%2B%2B\_Logo.svg.png 1.5x, //upload.wikimedia.org/wikipedia/commons/thumb/1/18/ISO\_C%2B%2B\_Logo.svg/260px-ISO\_C%2B%2B\_Logo.svg.png 2x" data-file-width="306" data-file-height="344" /></a></td></tr><tr><th><a href="/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F" title="Парадигма програмування">Парадигма</a>:</th><td>кілька: <a href="/wiki/%D0%9E%D0%B1%E2%80%99%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F" class="mw-redirect" title="Об’єктно-орієнтоване програмування">об’єктно-орієнтована</a>, <a href="/wiki/%D0%A3%D0%B7%D0%B0%D0%B3%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5%D0%BD%D0%B5\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F" title="Узагальнене програмування">узагальнена</a>, <a href="/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%BD%D0%B5\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F" title="Процедурне програмування">процедурна</a></td></tr><tr><th>Дата появи:</th><td><a href="/wiki/1983" title="1983">1983</a></td></tr><tr><th>Творці:</th><td><a href="/wiki/%D0%91%27%D1%8F%D1%80%D0%BD\_%D0%A1%D1%82%D1%80%D0%B0%D1%83%D1%81%D1%82%D1%80%D1%83%D0%BF" title="Б&#39;ярн Страуструп">Б'ярн Страуструп</a></td></tr><tr><th><a href="/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0\_%D1%82%D0%B8%D0%BF%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%97" title="Система типізації">Система типізації</a>:</th><td><a href="/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D0%B0\_%D1%82%D0%B8%D0%BF%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F" class="mw-redirect" title="Статична типізація">статична</a>, небезпечна, <a href="/w/index.php?title=%D0%9D%D0%BE%D0%BC%D1%96%D0%BD%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0\_%D1%82%D0%B8%D0%BF%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%97&amp;action=edit&amp;redlink=1" class="new" title="Номінативна система типізації (ще не написана)">номінативна</a></td></tr><tr><th>Основні реалізації:</th><td><a href="/wiki/GNU\_Compiler\_Collection" title="GNU Compiler Collection">G++</a>, <a href="/wiki/Microsoft\_Visual\_C%2B%2B" title="Microsoft Visual C++">Microsoft Visual C++</a>, <a href="/w/index.php?title=Borland\_C%2B%2B\_Builder&amp;action=edit&amp;redlink=1" class="new" title="Borland C++ Builder (ще не написана)">Borland C++ Builder</a>, <a href="/wiki/Clang" title="Clang">clang</a></td></tr><tr><th>Діалекти:</th><td><a href="/w/index.php?title=ISO/IEC\_14882&amp;action=edit&amp;redlink=1" class="new" title="ISO/IEC 14882 (ще не написана)">ISO/IEC C++</a> 1998, 2003, 2011, 2014, 2017, 2020(2a)</td></tr><tr><th>Під впливом від:</th><td><a href="/wiki/Smalltalk" title="Smalltalk">Smalltalk</a>,<a href="/wiki/%D0%A1%D1%96\_(%D0%BC%D0%BE%D0%B2%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)" class="mw-redirect" title="Сі (мова програмування)">C</a>, <a href="/wiki/Simula" title="Simula">Simula</a>, <a href="/wiki/Ada\_(%D0%BC%D0%BE%D0%B2%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)" class="mw-redirect" title="Ada (мова програмування)">Ada 83</a>, <a href="/w/index.php?title=ALGOL\_68&amp;action=edit&amp;redlink=1" class="new" title="ALGOL 68 (ще не написана)">ALGOL 68</a>, <a href="/wiki/CLU\_(%D0%BC%D0%BE%D0%B2%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)" title="CLU (мова програмування)">CLU</a>, <a href="/wiki/ML\_(%D0%BC%D0%BE%D0%B2%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)" class="mw-redirect" title="ML (мова програмування)">ML</a></td></tr><tr><th>Вплинула на:</th><td><a href="/wiki/Ada\_(%D0%BC%D0%BE%D0%B2%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)" class="mw-redirect" title="Ada (мова програмування)">Ada 95</a>, <a href="/wiki/C\_Sharp" title="C Sharp">C#</a>, <a href="/wiki/Java\_(%D0%BC%D0%BE%D0%B2%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)" class="mw-redirect" title="Java (мова програмування)">Java</a>, <a href="/wiki/PHP" title="PHP">PHP</a>, <a href="/wiki/D\_(%D0%BC%D0%BE%D0%B2%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)" title="D (мова програмування)">D</a>, <a href="/w/index.php?title=Aikido\_(%D0%BC%D0%BE%D0%B2%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)&amp;action=edit&amp;redlink=1" class="new" title="Aikido (мова програмування) (ще не написана)">Aikido</a>, <a href="/w/index.php?title=Dao\_(%D0%BC%D0%BE%D0%B2%D0%B0\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)&amp;action=edit&amp;redlink=1" class="new" title="Dao (мова програмування) (ще не написана)">Dao</a></td></tr><tr><th><a href="/wiki/%D0%A1%D0%B0%D0%B9%D1%82" class="mw-redirect" title="Сайт">Сторінка інтернет</a>:</th><td><a rel="nofollow" class="external text" href="https://isocpp.org/">isocpp.org</a></td></tr>  
 <tr><td></td></tr></tbody></table> -->  
 <section class="right">  
 <h2 id="t1">Скорочена довідка</h2>  
 <div class="line"></div>  
 <p><img src="../image/logo.png" alt="Логотип" class="logo"></p>  
 <div class="content">  
 <p class="content\_">C++ (Сі-плюс-плюс) — мова програмування високого[1][2] рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом (англ. Bjarne Stroustrup) в AT&T Bell Laboratories (Мюррей-Хілл, Нью-Джерсі) 1979 року та початково отримала назву «Сі з класами». Згодом Страуструп перейменував мову на C++ у 1983 р. Базується на мові С. Вперше описана стандартом ISO/IEC 14882:1998, найбільш актуальним же є стандарт ISO/IEC 14882:2014.  
 У 1990-х роках С++ стала однією з найуживаніших мов програмування загального призначення. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм, наприклад, відеоігор. С++ суттєво вплинула на інші популярні сьогодні мови програмування: С# та Java.</p>  
 </div>  
 </section>  
 <section class="left">  
 <h2 id="t2">Особливості</h2>  
 <div class="line"></div>  
 <div class="features">  
 <p>При створенні С++ прагнули зберегти сумісність з мовою С. Більшість програм на С справно працюватимуть і з компілятором С++. С++ має синтаксис, заснований на синтаксисі С.</p>  
 <p>Нововведеннями С++ порівняно з С є:</p>  
 <ul>  
 <li>підтримка об'єктно-орієнтованого програмування черезкласи;</li>  
 <li>підтримка узагальненого програмування через шаблони;</li>  
 <li>доповнення до стандартної бібліотеки;</li>  
 <li>додаткові типи даних;</li>  
 <li>обробка винятків;</li>  
 <li>простори імен;</li>  
 <li>вбудовані функції;</li>  
 <li>перевантаження операторів;</li>  
 <li>перевантаження імен функцій;</li>  
 <li>посилання і оператори управління вільно розподіленою пам'яттю.</li>  
 </ul>  
 <p>У 1998 році ратифіковано міжнародний стандарт мови С++: ISO/IEC 14882 «Standard for the C++ Programming Language». Поточна версія цього стандарту&#160;— ISO/IEC 14882:2011.  
 </p>  
 </div>  
 </section>  
 <section class="left">  
 <h2 id="t3">Приклад програми</h2>  
 <div class="line"></div>  
 <div class="example">  
 <p>Приклад програми «Hello, world!»</p>  
 <p>Нижче наведено приклад простої програми на С++, яка виводить на стандартний потік виводу рядок Hello, world!.</p>  
 <code>  
 <pre>  
 #include  
  
 using namespace std;  
  
 int main() {  
 cout << "Hello, world!" << endl;  
 return 0;  
 }  
 </pre>  
 </code>  
 <p>Приклад програми зі змінною</p>  
 <p>Після запуску програми на екрані з'явиться повідомлення How old are you?, а курсор буде розміщений в наступному рядку. Там потрібно ввести цілочислове значення та натиснути клавішу Enter. Приклад: вводимо число 25, тоді з'явиться повідомлення You are 25 years old.</p>  
 <code>  
 <pre>  
 #include  
  
 using namespace std;  
  
 int main() {  
 int age; // Змінна для запису віку  
 cout << "How old are You?\n"; // Скільки вам років?  
 cin >> age; // Зчитування віку  
 cout << "You are " << age << " years old\n"; // Результат  
 return 0;  
 }  
 </pre>  
 </code>  
 </div>  
 </section>  
 <section class="left">  
 <h2 id="t4">Хронологія розвитку</h2>  
 <div class="line"></div>  
 <div class="chronology">  
 <p>Історія розвитку мови програмування С++ містить такі ключові події:</p>  
 <ul>  
 <li>квітень 1979 — початок роботи над Ci з класами (C with Classes)</li>  
 <li>жовтень 1979 — робоча версія Ci з класами (Cpre)</li>  
 <li>серпень 1983 — C++ вперше використовується в Bell Labs</li>  
 <li>1984 — назва C++</li>  
 <li>лютий 1985 — перший зовнішній випуск C++ — Cfront Relese E (Educational — випуск для навчальних закладів)</li>  
 <li>жовтень 1985 — перший комерційний випуск — Cfront 1.0</li>  
 <li>лютий 1987 — Cfront 1.2</li>  
 <li>грудень 1987 — перший випуск GNU C++ (1.13)</li>  
 <li>1988 — Перші випуски Oregon Software C++ і Zortech C++</li>  
 <li>червень 1989 — Cfront 2.0</li>  
 <li>1989 — книга «The Annotated C++ Reference Manual» (ARM); Засновано комітет ANSI C++</li>  
 <li>1990 — перша технічна зустріч комітету ANSI C++; прийнято шаблони (templates), виняткові ситуації (exceptions); перший випуск Borland C++</li>  
 <li>1991 — Перша зустріч ISO; Cfront 3.0 (з шаблонами); книга «The C++ Programming Language» (2-га редакція)</li>  
 <li>1992 — Перші випуски IBM, DEC, Microsoft C++</li>  
 <li>1993 — RTTI (Run-time type identification — визначення типу під час виконання) прийнято; простори назв (namespaces) і string (шаблонний за символьним типом) прийнято</li>  
 <li>1994 — прийнято STL</li>  
 <li>1996 — прийнято export</li>  
 <li>1997 — остаточне голосування комітету за завершений стандарт</li>  
 <li>1998 — ратифіковано стандарт ISO C++ (так званий C++98)</li>  
 <li>2003 — технічні поправки до стандарту; початок роботи над C++0x</li>  
 <li>2005 — перше голосування за можливості C++0x; auto, static\_assert, rvalue references прийняті в загальному</li>  
 <li>2006 — Перше офіційне голосування з C++0x</li>  
 </ul>  
 </div>  
 </section>  
 <section class="left">  
 <h2 id="t5">Переваги та недоліки</h2>  
 <div class="line"></div>  
 <div class="advantages">  
 <h4>Переваги мови C++</h4>  
 <ul>  
 <li>Швидкодія. Швидкість роботи програм на С++ практично не поступається програмам на С, хоча програмісти отримали в свої руки нові можливості і нові засоби.</li>  
 <li>Масштабованість. На мові C++ розробляють програми для найрізноманітніших платформ і систем.</li>  
 <li>Можливість роботи на низькому рівні з пам'яттю, адресами, портами. (Що, при необережному використанні, може легко перетворитися на недолік.)</li>Можливість створення узагальнених алгоритмів для різних типів даних, їхня спеціалізація, і обчислення на етапі компіляції, з використанням шаблонів.  
 <li>Підтримуються різні стилі та технології програмування, включаючи традиційне директивне програмування, ООП, узагальнене програмування, метапрограмування (шаблони, макроси).</li>  
 </ul>  
 <h4>Недоліки мови C++</h4>  
 <ul>  
 <li>Наявність безліч можливостей, що порушують принципи типобезпеки приводить до того, що в С++ програми може легко закрастися важковловима помилка. Замість контролю з боку компілятора розробники вимушені дотримуватися вельми нетривіальних правил кодування. По суті, ці правила обмежують С++ рамками якоїсь безпечнішої підмови. Більшість проблем типобезпеки С++ успадкована від С, але важливу роль в цьому питанні грає і відмова автора мови від ідеї використовувати автоматичне управління пам'яттю (наприклад, збірку сміття). Так візитною карткою С++ стали вразливості типу «переповнювання буфера».</li>  
 <li>Погана підтримка модульності. Підключення інтерфейсу зовнішнього модуля через препроцесорну вставку заголовного файлу (#include) серйозно уповільнює компіляцію, при підключенні великої кількості модулів. Для усунення цього недоліку, багато компіляторів реалізують механізм прекомпіляції заголовних файлів (англ. Precompiled Headers).</li>  
 <li>Недостача інформації про типи даних під час компіляції (CTTI).</li>  
 <li>Мова C++ є складною для вивчення і для компіляції.</li>  
 <li>Деякі перетворення типів неінтуїтивні. Зокрема, операція над беззнаковим і знаковим числами видає беззнаковий результат.</li>  
 <li>Препроцесор С++ (успадкований від C) дуже примітивний. Це приводить з одного боку до того, що з його допомогою не можна (або важко) здійснювати деякі завдання метапрограмування, а з іншого, в наслідок своєї примітивності, він часто приводить до помилок і вимагає багато дій з обходу потенційних проблем. Деякі мови програмування (наприклад, Scheme і Nemerle) мають набагато могутніші і безпечніші системи метапрограмування (також звані макросами, але макроси С/С++ вони мало нагадують).</li>  
 <li>З кінця 1990-х в спільноті С++ набуло поширення так зване метапрограмування на базі шаблонів. По суті, воно використовує особливості шаблонів C++ в цілях реалізації на їхній базі інтерпретатора примітивної функціональної мови програмування, що виконується під час компіляції. Сама по собі ця можливість вельми приваблива, але, внаслідк вище згаданого, такий код вельми важко сприймати і зневаджувати. Мови Lisp/Scheme, Nemerle і деякі інші мають могутніші і водночас простіші для сприйняття підсистеми метапрограмування. Крім того, в мові D реалізована порівнянна за потужністю, але значно простіша в застосуванні підсистема шаблонного метапрограмування.</li>  
 <li>Хоча декларується, що С++ мультипарадигмена мова, реально в мові відсутня підтримка функціонального програмування. Частково, даний пропуск усувається різними бібліотеками (Loki, Boost) що використовують засоби метапрограмування для розширення мови функціональними конструкціями (наприклад, підтримкою лямбд/анонімних методів), але якість подібних рішень значно поступається якості вбудованих у функціональні мови рішень. Такі можливості функціональних мов, як зіставлення зі зразком взагалі украй складно емулювати засобами метапрограмування.</li>  
 </ul>  
 </div>  
 </section>  
 </main>  
 <footer>  
 <p>  
 <a href="../cv/index.html">Ivan Mysyk</a>  
 </p>  
 </footer>  
</body>  
</html>

Код CSS

body{  
 margin: 0;  
 padding: 0;  
 font-family: 'Roboto', sans-serif;  
}  
a{  
 text-decoration: none;  
}  
header {  
 border-bottom:5px solid #b99d61;  
 text-align:center;  
 background: #FAFAD2;  
 padding: 20px;  
}  
header h1{  
 font-size: 36px;  
}  
main {  
 margin:0 50px 0;  
 padding:0;  
}  
.widget {  
 padding: 20px 30px;  
 background: white;  
}  
.widget-title {  
 text-transform: uppercase;  
 letter-spacing: 2px;  
 color: #222;  
 font-size: 30px;  
 padding-left: 15px;  
 margin-bottom: 15px;  
}  
.widget-list {  
 padding: 0;  
 list-style: none;  
 display: flex;  
 justify-content: space-between;  
}  
.widget-list a {  
 text-decoration: none;  
 outline: none;  
 display: flex;  
 padding: 6px 10px;  
 letter-spacing: 1px;  
 font-size: 20px;  
 border: 2px solid #b99d61;   
}  
.widget-list a:hover {color: #b99d61;}  
.right{  
 float: right;  
 width: 20%;  
}  
.left{  
 float: left;  
 width: 75%;  
}  
.line{  
 border-top: 2px solid #b99d61;   
}  
.content{  
 display: flex;  
 flex-direction: column;  
}  
.content\_{  
 font-size: 16px;  
}  
.logo{  
 width: 250px;  
 border: 2px solid #b99d61;  
 padding: 20px 30px;  
}  
footer {   
 padding: 20px;  
 border-top:5px solid #b99d61;  
 clear: both;  
 background: #FAFAD2;  
}  
footer p{   
 font-size: 26px;  
 text-align: center;  
}

Контрольні запитання

1. **Чим відрізняються блочні та рядкові елементи?**

Блочні елементи можна представляти як прямокутні області на сторінці.

Рядковий елемент не має керованої висоти чи ширини і відображається поряд з іншими рядковими елементами.

1. **Перерахуйте три блочних та три рядкових тега.**

<p>, <ul>, <h1> - блочні

<b>, <i>, <ul> - рядкові

1. **Як значення block, inline та inline-block властивості display впливає на відображення елемента?**

Block - Елемент показується як блоковий. Застосування цього значення для вбудованих елементів, наприклад тега <span>, змушує його вести подібно блокам - відбувається перенесення рядків на початку і в кінці вмісту.

Inline - Елемент відображається як вбудований. Використання блокових тегів, таких як <div> і <p>, автоматично створює перенос і показує вміст цих тегів з нового рядка. Значення inline скасовує цю особливість, тому вміст блокових елементів починається з того місця, де закінчився попередній елемент.

Inline-block - Це значення генерує блоковий елемент, який обтікає іншими елементами веб-сторінки подібно вбудованому елементу. Фактично такий елемент по своїй дії схожий на вбудовані елементи (на кшталт тега <img>). При цьому його внутрішня частина форматується як блоковий елемент, а сам елемент - як вбудований.

1. **Що означає тег <div>?**

Елемент <div> є блоковим елементом і призначений для виділення фрагмента документа з метою зміни виду вмісту.

1. **Що означає тег <span>?**

Тег <span> призначений для визначення малих елементів документа. На відміну від блокових елементів, таких як <table>, <p> або <div>, за допомогою тега <span> можна виділити частину інформації всередині інших тегів і встановити для неї свій стиль.

1. **Які теги HTML5 призвані зменшити надмірне використання тега <div> для структурування сторінки?**

<section>, <header>, <footer>, <aside>

1. **Які властивості впливають на розмір місця, яке блок займає на сторінці?**

Padding, margin, border.

1. **Що таке padding?**

Порожній простір від контенту до внутрішнього краю меж елемента

1. **Що таке margin?**

Невидимий порожній простір від зовнішнього краю кордонів до сусіднього елемента.

1. **Що означає запис padding: 30px 50px 10px;?**

Верхній відступ 30px, лівий та правий 50px, нижній 10px.

1. **Що означає запис margin: 0 auto;?**
2. **Як буде виглядати блок з властивістю border-bottom: 3px dotted #ff0000;?**

Блок з нижнім кордоном в вигляді крапок товщиною 3px червоного кольору.

1. **На що впливає властивість box-sizing?**

На ширину та висоту елементу.

1. **Як задати ширину блока в абсолютних одиницях вимірювання та в процентах?**

Width: 50%, width: 50px

1. **Як обмежити розмір блока при динамічному розрахунку ширини?**
2. **Для чого використовують параметр overflow? У чому різниця між overflow, overflow-x та overflow-y?**

Якщо після завдання розмірів зміст блоку виявляється більшим за розміром, ніж може поміститись в блок, контент вийде за межі блоку. Для керування відображенням у такому випадку використовують властивість *overflow.*

1. **Поясніть використання властивості float.**

Щоб розмістити кілька блочних елементів поруч, використовують властивість *float*.

1. **Як відмінити обтікання контенту, задане за допомогою float?**

clear: both;

1. **Які значення може приймати властивість position?**

Використовується, щоб примусово змінювати положення елемента відносно інших елементів.

static, relative, absolute, fixed.

1. **Як впливають на розташування елемента властивості left, right, top, bottom при різних значеннях position?**

Static – не впливає

Relative - змінює позицію елемента і зрушує його в ту чи іншу сторону від первісного розташування

Absolute - На положення впливає значення властивості *position* батьківського елемента.

Fixed - прив'язується до зазначеної властивостями *left, top, right* і *bottom* точці на екрані і не змінює свого положення при прокручуванні сторінки.

1. **Що таке z-index?**

z-index керує розміщення елементів по осі z. Це властивість працює тільки для елементів, у яких значення *position* задано як absolute, fixed або relative.