 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №1**

з дисципліни «Алгоритмізація та програмування»

на тему: «Інтегроване середовище розробки програм Microsoft Visual Studio»

Варіант №17

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

доцент каф. БМК

к.т.н. Алхімова С.М.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2019

**Завдання:**

1. Розібратися з середовищем Microsoft Visual Studio 2010 та створити консольну програму за наведеним прикладом програмного коду:

#include <stdio.h>

#include <conio.h>

int main()

{

printf("\n\t Hello world!");

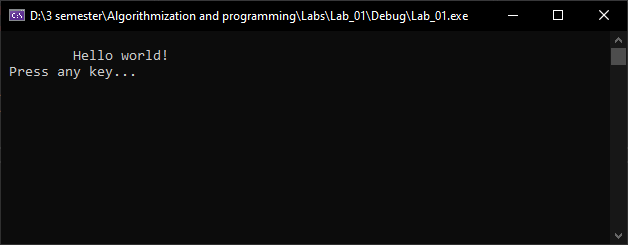
printf("\n Press any key...");

\_getch();

return 0;

}

1. Скомпілювати програму. В разі вірного виконання на екрані має запуститися програма, вікно якої матиме наступний вигляд:



**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

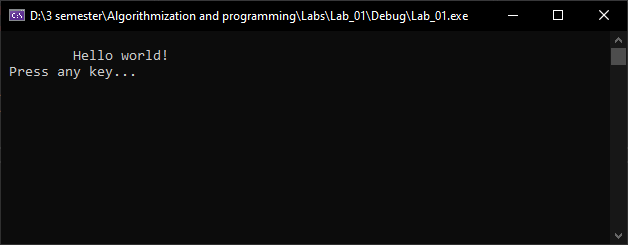
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Результати:**

****

**Контрольні питання:**

1. ***Що таке процес компіляції програми?***

Компіляція це процес перетворення комп'ютерної програми, написаної на мові программування у відповідний код на іншій мові програмування. Компілятор - це програмне забеспечення яке виконує це перетворення. Інколи цей процес називають "збирання" або "побудова", що як правило, означає бульше, ніж просто виконання компіляції, наприклад упакування його у двійковому форматі.

1. ***Що таке процес компанування програми?***

Компанування – приймання на вхід один або кілька об'єктних модулів і збирання їх в один виконуваний модуль.

Для зв'язування модулів компонувальник використовує таблиці символів, створені компілятором в кожному з об'єктних модулів. Ці таблиці можуть містити символи наступних типів:

* Визначені імена - функції і змінні, визначені в даному модулі та надаються для використання іншим модулям;
* Невизначені імена - функції і змінні, на які посилається модуль, але не визначає їх усередині себе;
* Локальні - можуть використовуватися всередині об'єктного файлу для спрощення процесу налаштування адреси.

1. ***Які системи називають інтегрованими середовищами розробки програм?***

Інтегроване середовище розробки – комплексне програмне рішення для розробки програмного забезпечення. Зазвичай, складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм. Більшість сучасних середовищ розробки мають можливість автодоповнення коду.

1. ***Перелічити основні головні риси комерційних варіантів Microsoft Visual Studio?***

Можливість працювати для підприємства, динамічна перевірка залежностей, діаграми архітектурного рівня, перевірка архітектури, клонування коду, інтеграція відладчика Code Map, аналіз дампа пам'яті .NET, відлагоджувач моментальних знімків, налагодження переходу за часом, динамічне модульне тестування, IntelliTest, імітації Microsoft (ізоляція модульних тестів), впроваджені збірки, Xamarin Inspector, Xamarin Profiler.

1. ***Процес створення нового проектного файлу в середовищі Microsoft Visual Studio.***
   1. Відкрити Visual Studio
   2. Натиснути новий проект
   3. Назвати його та повідомити його місцезнаходження
   4. Проектний файл створено!
2. ***Процес додавання нового файлу вихідного коду в середовищі Microsoft Visual Studio.***
   1. Відкрити проект
   2. У вкладці «Project» натиснути «Add new item»
   3. Вибрати назву та тип файлу
   4. Файл вихідного коду створено!
3. ***Як відредагувати ім’я файлів, які входять в проект?***
   1. Вибрати правою кнопкою миші файл
   2. Натиснути «Rename»
   3. Вибрати нову назву файлу
4. ***Призначення кожної з папок, що існує в проекті за замовчуванням.***

Папка Source Files призначена для файлів з вихідним програмним кодом, що мають розширення СРР (\*.срр).

Папка Header Files містить файли заголовків з розширенням Н (\*.h).

Папка Resource Files містить файли ресурсів, наприклад, зображення.

Папка External Dependencies відображує файли, які не були явно додані до проекту, але використовуються в файлах вихідного програмного коду, наприклад, підключені за допомогою директиви #include. Зазвичай в папці External Dependencies наявні файли заголовків стандартних бібліотек, що використовуються в проекті.

1. ***Які існують способи запуску програмного застосунку з інтегрованого середовища розробки програм Microsoft Visual Studio?***

Для того, щоб запустити програмний застосунок, існує декілька способів:

* Start Debugging (F5) - почати відлагодження, при цьому запускається програма під відладчиком;
* Start without Debugging (Ctrl+F5) - запустити без відлагодження, при цьому просто здійснюється запуск скомпільованої програми;
* Step Into (F11)- покрокове виконання з заходженням до тіла функції;
* Step Over (F10) - покрокове виконання без заходження у функцію;

1. ***Яким чином отримати контекстну довідку в інтегрованому середовищі розробки програм Microsoft Visual Studio?***

Достатньо вибрати частину коду, яка вас цікавить та натиснути «F1»