Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

КОНСПЕКТ

курс Увод в програмирането
за специалности Информатика, Компютърни науки и Софтуерно инженерство
зимен семестър 2016/2017 г.

1. Уводна лекция
Кратка история на развитието на програмирането.
Модели (императивен, ламбда смятане/функционално програмиране; формален извод/логическо програмиране).
Кратка история на езика C++.

2. Основи. Основни елементи на езика
Двоични числа и работа с двоични числа.
Побитови операции.
Основни елементи на езика: Коментари, идентификатори и запазени думи, основни типове, литерали и константи, променливи, оператори, изрази, преобразуване на типовете, типове дефинирани от потребителя. enum, tydef.pe
Вход и изход.

3. Инструкции (statements)
Инструкция-израз (expression statement)
Съставна инструкция (compound statement)
Инструкции за избор (selection statements)
Инструкции за цикъл (iteration statements)
Инструкции за преход (jump statements)
Блок схеми

4. Функции
Какво представляват функциите.
Дефиниция, декларация, прототип.
Подаване на параметри по стойност, указател и псевдоним.
Вътрешна реализация на извикване на функция чрез стека. Стекови рамки.

5. Рекурсия

6. Масиви

Какво представляват масивите.

Основни операции с масиви:

- Поддържане на размера на масива по-голям или равен на броя на съхранените в него елементи и запазване на този брой в отделна променлива;
- Добавяне на елемент (и изместване вдясно);
- Премахване на елемент (и изместване вляво).

Основни алгоритми за сортиране - пряка селекция и метод на мехурчето.

Двоично търсене.

Многомерни масиви.

Физическо представяне на масивите в паметта.

Работа с матрици.

7. Работа с паметта

Основни принципи. Стек и хийп.

Как стекът се използва при работа с функции и при дефиниране на автоматични променливи.

Указатели и работа с тях. Какво е указател, за какво се използва, примери с масиви и указатели, указателна аритметика.

Псевдоними: дефиниране и използване.

Указател към неизвестен тип (void*).

Указатели/псевдоними и константи.

Работа с динамичната памет. Оператори new и delete.

8. Работа с текст

Основи и обща информация - как се представя текстът в компютърните системи.

Кодировки. Single-byte vs multi-byte. Широки (wide) символи. Unicode. UTF-8.

Стандартно представяне на символните низове в C++.

Основни операции със символни низове - strlen, strcat, strcpy, strstr и др. Описание на тези операции и реализация.

Потенциални проблеми и рискове за сигурността свързани с тези операции. По-добри алтернативи (например вместо strcpy, използване на strncpy в GCC или strcpy_s във VC++).

Въвеждане на символни низове низове с потоците - особености, поведение на << и >>, get и getline.

9. Функции от по-висок ред

Отново за дефинирането на потребителски типове. Използване на using от C++11 вместо typedef.

Указатели към функции.

Функциите от по-висок ред като шаблони на изчислителни процеси. Функции accumulate (fold), map и filter
Връщане на функции като резултат.

10. Записи (struct)
Логическо описание и дефиниция.
Представяне в паметта. Подравняване (alignment) и запълване (padding).
Операции над записи.
Сложни типове данни: масив от записи, запис от записи.
Записи като параметри и резултат на функции.
Указатели и псевдоними на записи, оператори точка (.) и стрелка (->)
Рекурсивни записи, свързана верига.
Абстракция със структури от данни (СД).

11. Конвенции за оформяне на кода. Добри практики.
През семестъра тази тема се разглежда паралелно с останалите. При въвеждането на нови понятия ще бъде показвано как да организираме кода си и кои са добрите практики. В края на курса се очаква студентите да могат да организират и документират кода си съгласно добрите практики.

12. Работа със средата за разработка. Работа с Debugger.
Също тема, която се разглежда паралелно с останалите. При въвеждането на нови понятия ще бъде обсъждано и демонстрирано как средата за разработка ги поддържа и какви средства за debug предлага тя (например step-ване, показване на стойности на променливи в реално време, call stack, memory, watch и т.н.).
След успешно преминаване на курса, от студентите се очаква да владеят добре поне една среда за разработка и да използват поне на средно ниво инструментите за debug, които предлага тя.