1. **Які є етапи розробки програми мовою C++?**

Є кілька типових кроків розробки програми мовою C++:

* початковий код готується за зберігається за допомогою будь-якого текстового редактора;
* здійснюється *препроцесування* (preprocessing) вихідного коду і створюється одиниця трансляції; в більшості випадків препроцесування об'єднане з компіляцією;
* здійснюєтсья компіляція вихідного коду; якщо компіляція неможлива через наявність *помилок компіляції* (compiler errors) , виводиться список цих помилок; якщо синтаксичних помилок немає, створюється так званий *об'єктний модуль* - файл з машинним кодом, який містить посилання на функції, відсутні цьому модулі; найчастіше об'єктний модуль - це файл з розширенням.obj.
* здійснюється редагування невирішених зв'язків за допомогою так званого *компонувача* (linker) - спеціальної підпрограми, яка зв'язує декілька об'єктних модулів, а також додає скомпільовані функції зі стандартних бібліотек - файлів з розширенням lib; крім того, з коду видаляються функції, на які немає посилань; результат роботи компонувача - виконуваний файл (\*.exe) або бібліотека.

Під час роботи виникають *помилки часу виконання* (runtime errors), які відображаються в окремих діалогових вікнах, а також *логічні помилки* (logical errors, програма виконує неочікувані дії і ми не отримуємо результату). Виправлення цих помилок вимагає повторного редагування коду, компіляції, компоновки та завантаження скорегованої програми на виконання

1. **У чому різниця між компілятором і компонувальником?**

здійснюється редагування невирішених зв'язків за допомогою так званого компонувача (linker) - спеціальної підпрограми, яка зв'язує декілька об'єктних модулів, а також додає скомпільовані функції зі стандартних бібліотек - файлів з розширенням lib; крім того, з коду видаляються функції, на які немає посилань; результат роботи компонувача - виконуваний файл (\*.exe) або бібліотека.

1. **У чому різниця між коментарями стилю C і стилю C++?**

стиль C: між словами замість пропуску вживають символ підкреслення ("\_");

стиль C++: слова пишуть злито, починаючи друге, третє та інші слова з великої літери (camel notation).

у стилі C++ - починаючи з символів // компілятор проігнорує весь текст до кінця поточного рядка;

у стилі C - починаючи з символів /\* до символів \*/ у цьому або в наступних рядках весь текст буде проігноровано компілятором; всередині тексту коментаря не можуть зустрічатися пари /\* та \*/.

1. **Що таке препроцесор?**

Окрім безпосередньо тексту мовою C++, початковий код зазвичай також містить директиви препроцесору. Ці директиви починаються з символу # і їх обробляє спеціальна програма - препроцесор

1. **У чому різниця між знаковими і беззнаковими цілими?**

Целые числа int, short int, long int, char без определенного знака

1. **Що таке константа?**

*Константне значення* (неіменована константа) - це лексема, яка представляє певне числове або символьне значення, а також рядок символів.

*Константи цілого типу*записуються як послідовності десяткових цифр. Тип константи залежить від числа цифр у записі константи і може бути уточнений додаванням наприкінці константи букв L чи l (тип **long**), U або u (тип **unsigned**) або у сполученні. Цілі константи можуть записуватися у вісімковій системі числення, у цьому випадку першою цифрою повинна бути цифра 0, число може містити тільки цифри 0...7. Цілі константи можна записувати й у шістнадцятковій системі числення. У цьому випадку запис константи починається із символів 0x чи 0X. Для позначення цифр понад 9 використовуються латинські букви a, b, c, d, e та f (великі або маленькі).

*Константи символьного типу***char** беруть в одиночні лапки (апострофи), значення константи визначається знаком з поточного набору символів або цілою константою, якій передує зворотна коса риска (символ із заданим кодом). Є ряд спеціальних символів, що можуть використовуватись як значення константи типу **char** (такі подвійні символи називаються керуючими послідовностями, або escape-послідовностями):

*Константи дійсних типів*можуть записуватись у формі з плаваючою крапкою або в експонентному форматі і за умовчанням мають тип **double** (число з плаваючою крапкою подвійної точності). За необхідності тип константи можна уточнити, записавши наприкінці суфікс f чи F для типу **float**, суфікс l чи L для типу **long double**. Можна також явно використовувати суфікси d і Dдля **double**. Наприклад:

1.5f // 1.5 типу float

2.5E-2d // 0.25 типу double

*Константа-рядок*(літерал, string literal) складається із символів, які беруть у подвійні лапки. Наприклад:

"Це рядок"

1. **Як визначити шістнадцяткову константу?**
2. **Що таке керуюча послідовність?**
3. **Що таке літерал?**

Константа-рядок (літерал, string literal) складається із символів, які беруть у подвійні лапки. Наприклад:

"Це рядок"

1. **Що таке змінна?**

Змінна – це іменована область пам'яті, до якої є доступ у програмі. Кожна змінна має певний тип. Зі змінною пов'язується її значення, яке зберігається в певному місці пам'яті (rvalue – праве значення, у присвоюванні стоїть праворуч), і її місце розташування, тобто адреса в пам'яті, за якою зберігається її значення (lvalue – ліве значення).

1. **Як визначити змінну?**

Визначення змінної викликає виділення пам'яті. Визначення задає ім'я змінної та її тип і закінчується крапкою з комою:

1. **Що таке іменована константа?**

Для подання константних величин можуть використовуватися так звані іменовані константи (symbolic constants). Значення іменованої константи не можна змінювати в програмі. Приклад опису константи:

const int yearOfBirth = 1901;

1. **Для чого використовують операцію sizeof?**

Унарна операція sizeof повертає розмір операнда у байтах. Операнд операції sizeof може бути одним з таких:

ім'я типу; щоб використовувати sizeof з ім'ям типу, ім'я має бути взяте в круглі дужки;

вираз; у цьому випадку операнд може бути вказаний з або без дужок; вираз не обчислюється.

1. **Що таке вираз?**

Вираз - це конструкція мови, яка складається з однієї або декількох операцій і може приводити до отримання певного значення під час виконання програми.

1. **Що таке операція?**

Операція це атомарний (неподільний) вираз. Об'єктами операцій є операнди. Залежно від кількості операндів виділяють унарні операції (один операнд), бінарні операції (два операнди) і тернарні операції (три операнди).

1. **У чому різниця між операціями присвоєння та складеного присвоєння?**
2. **У чому різниця між префіксними і постфіксними операціями інкременту і декременту?**
3. **Який тип операції відношення?**
4. **Які є логічні операції?**
5. **Що таке операція "кома"?**

Операція "кома" дозволяє замість одного виразу записувати кілька виразів, розділених комами. Ці вирази обчислюються зліва направо. Типом і значенням результату є тип і значення правого (останнього) виразу. Наприклад, запис

Наприклад, запис

x = (i = 0, j = i + 4, k = j);

еквівалентний запису

i = 0; j = i + 4; k = j; x = k;

Дужки в цьому прикладі є обов'язковими.

1. **Що таке умовна операція?**

Умовна операція (?:) - це єдина тернарна операція C++ (операція, яка приймає три операнди)

1. **Що таке інструкція (твердження, statement)?**

У мові C++ інструкція (твердження, statement, іноді оператор) є мінімальною одиницею програмного коду. Вирази та операції входять в інструкції. Послідовність інструкцій реалізує певний алгоритм - лінійний, з розгалуженням або циклічний.

Порожня інструкція (null statement) - це просто крапка з комою. Порожню інструкцію вживають, коли інструкція необхідна синтаксично, але не треба виконувати ніяких дій.

1. **Що таке інструкція-вираз?**

Усі вирази, які завершуються крапкою з комою, є інструкціями - інструкціями-виразами (expression statements). Однією з найбільш інструкцій є інструкція з виразом присвоювання:

a = x + 1;

1. **Для чого використовують складену інструкцію (блок)?**

Складена інструкція – це послідовність інструкції, укладена у фігурні дужки {}. Складену інструкцію часто іменують блоком. Після фігурної дужки, яка закриває блок, крапка з комою не ставиться. Синтаксично блок може розглядатися як окрема інструкція. Наприклад:

{

temp = a;

a = b;

b = temp;

}

1. **Опишіть синтаксис інструкції if.**
2. **Опишіть синтаксис інструкції switch.**
3. **Як використовувати інструкцію break всередині інструкції switch?**
4. **Для чого використовують інструкцію default?**
5. **У чому різниця між циклом з передумовою і циклом з постумовою?**

Цикл з передумовою будується за схемою

while (вираз-умова)

інструкція

Під час кожного повторення циклу (ітерації) обчислюється вираз-умова і якщо значення цього виразу не дорівнює нулю, виконується інструкція - тіло циклу.

Цикл з постумовою будується за схемою

do

інструкція

while (вираз-умова);

У цьому варіанті вираз-умова обчислюється і перевіряється після кожного повторення інструкції - тіла циклу, цикл повторюється, поки умова виконується. Тіло циклу в циклі з постумовою виконується хоча б один раз.

1. **У чому переваги інструкції for?**
2. **У чому різниця між інструкціями break і continue?**