**Практична робота №5. Тематичне опитування**

1. Що включає в себе алфавіт мови С++?
2. Що таке лексеми, як вони складаються, що до них відносять?
3. Чим визначаються межі лексем?
4. Дайте визначення ідентифікатору, яким чином він утворюється?
5. Чому дорівнює вираз int c = 10 >= 4; ?
6. Що називають ключовими словами?
7. Які символи можуть бути роздільниками лексем?
8. Які коментарі використовує мова C++? Надайте поради до їх створення, наведіть приклади.
9. Що являє собою структура програми на C++? Наведіть основні вимоги, які слід ураховувати при створенні програм мовою C++.
10. Які основні типи даних має мова С++ та як вони ідентифікуються ?
11. Які складені типи даних має мова С++?
12. Для чого використовуються специфікатори типу даних? Назвіть їх
13. Що таке змінна і як здійснюється її опис та визначення?
14. Які дії виконуються компілятором при обробці опису змінної?
15. Що таке «область дії ідентифікатора»?
16. Що таке «клас пам’яті»?
17. Чим локальна змінна відрізняється від глобальної?
18. Чи можуть бути в програмі локальна та глобальна змінні з однаковим іменем?
19. Чому дорівнює вираз int c = 10 != 4;?
20. В чому полягає операція**ініціювання змінної*?***
21. Які константи налічує С++? Охарактеризуйте їх застосування.
22. Що таке пріоритет операції? Наведіть приклади для арифметичних та логічних операцій
23. **Як визначається експоненціальна форма** дійсного числа?
24. Що називають керуючими або escape-послідовностями?
25. Які операції присвоювання та операції відношення налічує С++?
26. Що реалізують логічні операції та операції обробки окремих бітів?
27. Які операції над покажчиками має С++?
28. Що таке **Сумісність типів за присвоюванням ?**.
29. Які додаткові операції має С++?
30. Чому дорівнює вираз int d = 4 != 4; ?
31. Що таке перетворенням типу?
32. Які правила запису ідентифікаторів?
33. Яка відмінність змінної від константи?
34. Які є форми запису дійсних чисел у мові С++?
35. Як здійснюється виведення на екран в мові С++?
36. Які є форми запису операцій інкременту та декременту? Яка між ними різниця?
37. Що таке пріоритет виконання операцій? Перерахуйте відомі Вам операції мови С++ в порядку зростання старшинства?
38. Що таке бібліотека стандартних функцій? Для чого вони використовуються?
39. Яка різниця між виразом та інструкцією присвоювання?
40. Які обов'язкові частини повинна мати програма мовою C++?
41. Як забезпечується організація введення даних на С++?
42. Як здійснюється виведення результатів на С++?
43. Як реалізується потокове введення даних «**сіn>>**»? Наведiть приклади.
44. Охарактеризуйте основні аспекти використання потокового виведення даних «**соut<<**».
45. Що таке форматоване введення-виведення даних?
46. Поясніть правила застосування функції **scanf**.
47. Як здійснює роботу функція **printf**?
48. Для чого слугують заголовні файли?
49. Як компілятор шукає розташування заголовних файлів?
50. Які маніпулятори використовуються для завдання довжини поля, дробової частини числа?
51. Який заголовний файл використовується для маніпуляторів завдання довжини поля, дробової частини числа?
52. Для чого ми записуємо **using namespace std;?**
53. Для чого ми записуємо
54. Визначте основні частини типової структури програми на С++.
55. Що таке макровизначення? Наведіть приклади.
56. Що може міститися в зовнішніх оголошеннях?
57. Які значення може містити умова, що перевіряється?
58. Як визначається результат операція **not?**
59. Як визначається результат операція **and?**
60. Як визначається результат операція **or?**
61. Поясніть, що представляє собою логічний вираз.
62. Як визначається тернарна операція?
63. Для чого в програмах використовуються коментарі?
64. Чи можна створити змінну або константу з іменем **sqrt**,**main**?
65. Чи можна створити змінну або константу з іменем **int**, **const**?
66. В чому полягає трансляція програми?
67. В чому полягає компіляція програми?
68. В чому полягає інтерпретація програми?
69. Що виконує директива препроцесора #define?
70. Чи ставиться в кінці директиви препроцесора ;?

**Розглянути**

Арифметичний тип вважається **більшим** за інший арифметичний тип, якщо для зображення його значень потрібно більше байтів або він має більше максимальне значення. Уникайте присвоювань значень більшого цілого типу змінним меншого цілого типу, оскільки це може мати непередбачувані наслідки. Зокрема, уникайте перетворення значень беззнакових цілих типів на значення знакових типів.

Пріоритет порівнянь нижчий за пріоритет **+**, **-**, **<<** і вищий за пріоритет **&&**, **||**. Оператори **>, >=**, **<**, **<=** мають вищий пріоритет ніж **==** та **!=**, а **&&** – вищий ніж **||**.

Компілятори С++ забезпечують так зване **ледаче**11, або **скорочене**, обчислення булевих операцій **&&** та **||**. Спочатку обчислюється їх перший операнд. Якщо для операції **&&** це **false**, то другий операнд обчислювати не треба, адже результатом усе одно буде **false**. Аналогічно, якщо перший операнд операції **||** має значення **true**, то другий операнд не потрібен.

**Приклад**. У виразі **(2\*2==5)&&(323345%2209==37)** обчислюється тільки **2\*2==5** (хибність), а у виразі **(2\*2==4)||(323345%2209==37)** – тільки **2\*2==4** (істина).

**Вправи**

1. Поясніть, у чому різниця між **314**. та **314**

2. Поясніть, у чому різниця між **12.151** та **"12.151"**.

**Розглянути**

**int d;** //1 — глобальна змінна **d**

**int main()**

**{**

**int b;** //2 — локальна змінна **b**

**extern int y;** //3 — змінна **у** визначена в іншому місці програми

**static int s;** //4 — локальна статична змінна **s**

**d = 1;** //5 — присвоювання значення глобальній змінній

**int d;**   //6 — локальна змінна **d**

**d = 10;** //7 — присвоювання значення локальній змінній

**::d = 3;**  //8 — присвоювання значення глобальній змінній

**return 0;**

**}**

int a = 8;

cout << ++a; // На екрані 9

cout << a++; // На екрані 9

cout << a; // На екрані 10

Якщо крім суфіксної форми інкремента чи декремента є інші оператори, то спершу виконається інша команда (в залежності від оператора) і тільки після цього буде виконане додавання/віднімання 1. **cout << a++;** Спершу буде виконано операцію виводу і на екран виведеться число і тільки після того буде виконана операція декремента.

**Що буде виведено за програмою?**

**#include <iostream>**

**using namespace std;**

**int main() {**

**cout<<'1'; cout<<'\n';**

**cout<<'Y'; cout<<'e';**

**cout<<'s'; cout<<endl;**

**system("pause"); return 0;**

**}**

**Вправа 3**. Що буде виведено на екран, якщо під час виконання програми введено символи **15 n** та натиснуто на клавішу **Enter**?

**#include <iostream>**

**using namespace std;**

**int main() {**

**int n; char a;**

**cin >> n >> a;**

**cout<<"n="<<n<<' '<<"a="<<a<<endl;**

**system("pause"); return 0;**

**}**

**Розглянути**

1. Значенням виразу **4+7/5** є **5**, оскільки спочатку обчислюється **7/5** із результатом **1**, а потім **4+1** із результатом **5**.
2. Значенням виразу **(4+7)/5** є **2**, оскільки спочатку обчислюється операція в дужках **(4+7)** – її значення 11, а потім **11/5** із результатом **2**.
3. У виразі **sizeof 2.0+4** обчислюється **sizeof** із результатом **8**, потім додається **4**.
4. Значення виразу **4-3-2** дорівнює **-1**, оскільки спочатку обчислюється **4-3**, тобто **1**, а потім **1-2**;
5. у виразі **2\*7%8** спочатку обчислюється **2\*7** (це **14**), потім **14%8**, тобто **6**.
6. Корінь із невід'ємного дискримінанта квадратного рівняння з дійсними коефіцієнтами **a, b, c** можна обчислити виразом **sqrt(b\*b–4\*a\*c)**, а дійсні корені рівняння – виразами**(-bsqrt(b\*b–4\*a\*c))/(2\*a)** та **(-b+sqrt(b\*b–4\*a\*c))/(2\*a)**.
7. Дужки в знаменнику обов'язкові. Якщо їх не записати, то відбудеться не ділення, а множення на **a**.
8. Вираз **pow(b\*b–4\*a\*c,0.5)** позначає обчислення квадратного кореня з **b\*b–4\*a\*c**, вираз **pow(b,1.0/3.0)** – обчислення кубічного кореня з **b**, а обидва вирази **pow(2.0, 5)** та **pow(2, 5.0)** – піднесення дійсного числа **2.0** до степеня **5**. Зверніть увагу: вираз **pow(2, 5)** із двома цілими аргументами є помилковим.
9. Значенням **log10(2.0)** є (наближено) 0.30103, значенням **log(1)** – дійсне 0.
10. Значенням **fabs(-2.0)** є дійсне **2.0**, значенням **abs(-2)** – ціле **2**.
11. Вираз **int(2.8)** має значення 2, **double(-2)/3** – значення **- 0.666…**, а **double(-2/3)** – значення **0.0**

**Вправи**

4. Що буде виведено на екран за такими інструкціями?

**cout<<9/5<<' '<<-9/5<<' '<<9/-5<<' '<<-9/-5<<endl;**

**cout<<9%5<<' '<<-9%5<<' '<<9%-5<<' '<<-9%-5<<endl;**

**cout<<9./5.<<endl;**

5. Що буде виведено на екран за такими інструкціями?

**cout<<7/3<<' '<<1/6<<endl;**

**cout<<7./3.<<' '<<1./6.<<endl;**

6. Припустимо, що значення дійсної змінної **length** відповідає довжині будівлі в міліметрах. Написати вираз, що задає довжину будівлі в метрах.

7. Нехай значення цілої змінної **sizeOfFile** задає розмір файлу в байтах. Написати вираз, значенням якого є розмір файлу в Кбайтах.

8 . Нехай **v** – ім'я цілої змінної з невід'ємним значенням. Написати вираз, який обчислює: а) значення молодшої десяткової цифри числа **v**; б) значення молодшої двійкової цифри числа **v**.

9. Що буде виведено на екран за такими інструкціями?

**cout<<1+4/2<<' '<<(1+4)/2<<endl;**

**cout<<2\*4%7<<' '<<2\*(4%7)<<endl;**

**cout<<51/6%7<<' '<<51/(6%7)<<endl;**

**cout<<2\*(7%8)<<' '<<12/6%8<<' '<<5-3%2<<endl;**

10. Що буде виведено на екран за такими інструкціями?

**cout<<4\*6/8<<' '<<4/8\*6<<endl;**

11. Що буде виведено на екран за такими інструкціями?

**cout<<(-3+5)\*(2%7/3+4\*2)<<endl;**

12. Що буде виведено за такими інструкціями?

а) **int a=2, b=3;**

**cout<<"a\*b="<<a\*b<<";\n\*\*\*"<<endl;**

б) **int a=3, b; cout<<a\*a<<' '<<a+4<<'\n';**

13. Яке із за коментованих значень не буде давати помилку компіляції?

**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**int main(int argc, char\* argv[])**

**{ float fex;**

**fex=-123;**

**//fex=abs(fabs);**

**//fex=fabs(-123);**

**//fex=abs(-123);**

**cout << fex << endl;**

**system("pause");**

**return 0;**

**}**

**Приклади перетворення типів**

1. Якщо змінній **double x** присвоюється **1**, то її значенням стає дійсне **1.0**.

2. Якщо змінній **int a** присвоюється **1.9**, то її значенням стає **1**, а якщо **-1.9**, то **-1**.

**Приклад**. Вирази **(2<3)&&(2<4)**, **(0>1)||(1>0)** та **!('a'=='x')** мають значення **true**, вирази **(2<-2)&&true** та **!(5==5)** – значення **false**.

**Вправи**

14. У чому відмінність виразів **a=3** та **a==3**?

15. Обчислити значення виразів а) **1.5<7**; б) **'3'<'a'**; в) **'Z'<'a'**; г) **false<true**; д) **false<-1**? Які неявні перетворення типів виконуються під час їх обчислення?

16. Обчислити значення виразу: а) **!true==0;** б)**!false||(false==1)**; в) **(2>1)&&!true**.

17. Обчислити значення виразу **6<3&&7<5||3==5**.

**Приклади**

1. Ураховуючи пріоритети операцій, вираз **(k>0)&&(d>0)** можна записати як **k>0&&d>0**, але перша форма запису сприймається людиною краще.

2. Математичну умову *x*<*y*<*z* можна записати мовою C++ як **(x<y)&&(y<z)**. Запис **x<y<z** виражає зовсім іншу умову, а саме **(x<y)<z**. Обчислимо вираз **(-3<-2)<-1**. Значенням виразу **(-3<- 2)** є **true**. Далі обчислюємо **true<-1**, починаючи зі зведення аргументів порівняння до типу **int**, тобто в кінці обчислюється **1<-1** і його значенням є **false**. Проте *з погляду математики* запис -3<-2<-1 істинний.

3. Математичну умову *x*=*y*=*z* того, що числа *x*, *y*, *z* попарно рівні між собою, можна записати мовою C++ як **(x==y) && (y==z).** Для дійсних змінних **x**, **y**, **z** вираз **x=y=z** еквівалентний виразу **x=(y=z)** і задає присвоювання значення змінної **z** змінним **y** та **x**; його результатом буде значення змінної **z** – деяке дійсне число.

**Приклади**

1. Умовою того, що число *x* належить проміжку [*a*; *b*], є математичний вираз *a* *x* *b*. Виразимо її так: **(a<=x)&&(x<=b).**

Аналогічно умову того, що значення змінної **char c** є десятковою цифрою, можна записати як **('0'<=c)&&(c<='9')**.

2. Припустимо, що значення числових змінних **a**, **b**, **c** зображують довжини відрізків. Умовою того, що з відрізків можна утворити трикутник, є система нерівностей:

*a* > 0, *b* > 0, *c* > 0, *a*+*b* > *c*, *a*+*c* > *b*, *b*+*c* > *a*.

Цій системі відповідає вираз (дужки не обов'язкові, але додають наочності)

**(a>0)&&(b>0)&&(c>0)&&(a+b>c)&&(a+c>b)&&(b+c>a)**