

1. Які два файли звичайно визначаються для класу?

- Header and Main
- Header and Body
- Header and Source
- Class and Source

2. Для чого використовується файл заголовок?

- Для оголошення змінних та прототипів функцій
- Розміщення вихідного коду функції
- Ні для чого
- Тільки для розміщення конструкторів

3. Створіть функцію під назвою "test", що належить до класу, який називається "demo".

```
class demo {  
    test();  
}
```

4. Щоб використовувати клас, що слід включити до main.cpp?

- Файл заголовка класу
- Сpp-файл класу
- Обидва, заголовок та початковий файли

5. Яке твердження про конструктори та деструктори C++ є правильним?

- Конструктор працює, коли об'єкт видалено
- Конструктор працює, коли створюється об'єкт; деструктор працює при його видаленні
- Деструктор працює під час створення об'єкта

6. Оголосіть деструктор класу MyClass.

```
class MyClass {  
public:  
    MyClass();  
    _____;  
};
```

7. Визначте деструктор класу "MyClass", який друкує на екрані some text у його тілі.

```
MyClass:: _____ {  
    _____ "some text";  
}
```

8. Які з наведених тверджень справедливі щодо деструкторів?

- ☐ Деструктор завжди повертає тип double
- ☐ Деструктор не має імені
- ☐ Деструктора не має параметрів
- ☐ Деструктор не можна перевантажувати

9. Деструктори.

- ☐ ... викликається при видаленні об'єкта
- ☐ ... викликається, лише якщо конструктор не викликається
- ☐ ... змінні члена класу

10. Яка мета директив #ifndef #define у файлах заголовків?

- ☐ Вони запобігають включенню заголовкового файлу більше одного разу в один файл
- ☐ Вони необхідні для простору імен std
- ☐ Змушують програми працювати швидше

11. Створіть прототип функції myPrint при оголошенні класу Sally

```
_____ Sally {  
    _____:  
    Sally();  
    void _____();  
};
```

12. Оголосіть об'єкт класу Sally в функції main та зробіть виклик функції-члена myPrint використовуючи оператор точка (.).

```
int main() {  
    _____ obj;  
    _____ . _____;  
}
```

13. Оголосіть вказівник на obj

```
Sally obj;  
Sally _ sallyPtr = _ obj;
```

14. Додайте пропущений оператор доступу, щоб викликати функцію myPrint() через вказівник sallyPtr

```
Sally obj;  
Sally *sallyPtr = &obj;  
sallyPtr _____ myPrint();
```