

CS205 C/CPP Design Quiz

Note:

- Quiz has the following types of questions:
 - judgment questions (Give a statement, judge true or false)
 - Single choice/multiple choice questions (Give questions and options and select one or more of the correct items)
 - blank filling questions (Fill in a simple answer to the question)
- This material may be helpful for your class quizzes and final exams, so please keep it well.
- The producer of this material is **Maystern**.

1. The source code of program is as follows:

```
1 | int main(int arc, char *argv[]) {  
2 |     ...  
3 | }
```

The source code has been compiled to program **hello** , when you run it as follows, what **argc** will be ?

```
1 | ./hello I LOVE CPP
```

2. The following code is the **prototype/declaration** of function **mul** .

```
1 | int mul(int a, int b) {  
2 |     return a * b;  
3 | }
```

3. The data type of the variable **PI** is **double** .

```
1 | #define PI 3.14
```

4. When a function prototype is declared in the header file you create, you do NOT need to define it in a CPP file.

5. IDE (Integrated development environment) is a powerful compiler with many useful features.

6. If you get 75 for one of your projects, which situation should most likely be?

- A. Finish all tasks almost perfectly
- B. Finish all tasks well
- C. Correct
- D. Finish all tasks

7. What's the output of the source code?

```
1 | signed char c1 = 127;  
2 | signed char c2 = 1;  
3 | int csum = c1 + c2;  
4 | cout << csum;
```

8. `sizeof()` is a function and can yield the size in bytes of a type.

```
1 | int int_num = 10;
2 | size_t len = sizeof(int_num);
```

9. `size_t` is an unsigned integer type.

10. What's the value of the variable `num1`, `num2`, `num3` ?

```
1 | int num1 = 23 / 4 * 4;
2 | int num2 = 23 / 4 * 4.;
3 | int num3 = 23 / 4. * 4;
```

11. In the following code, since the variable `num` is not initialized explicitly, it will be initialized to 0 automatically.

```
1 | int num;
2 | cout << num << endl;
```

12. `auto` is a placeholder type specifier in C++11. What is the value of the variable `val` in the following code?

```
1 | auto val = 2 / 3;
2 | val = 3.14 * 2;
```

13. What's the output of the following source code ?

```
1 | int x = 100;
2 | int y = 30;
3 | x += (y -= 10);
4 | cout << x;
```

14. What's the output of the following source code ?

```
1 | int x = 100;
2 | int y = x++;
3 | cout << y;
```

15. What's the output of the following source code ?

```
1 | int x = 100;
2 | int y = (x = 200);
3 | cout << y;
```

16. What's the output of the following source code ?

```
1 | int a = 2, b = 0;
2 | if (a || b++) {
3 |     cout << b;
4 | }
```

17. What's the output of the following source code ?

```

1 | int i = 0;
2 | for (i = 1; i < 100; i += 2) {
3 |     // something
4 | }
5 | cout << i

```

18. What's the output of the source code ?

```

1 | int x = 5;
2 | do {
3 |     x = 0;
4 | } while (x);
5 | cout << x;

```

19. The following source code (empty in the parentheses) can be compiled successfully.

```

1 | for () {
2 |     // some lines here
3 | }

```

20. What is the output of the following code ?

- A. Key 4
- B. Key 5
- C. Undefined key.

```

1 | int num = 5;
2 | switch (num) {
3 |     case '4':
4 |         cout << "Key 4" << endl;
5 |     case '5':
6 |         cout << "Key 5" << endl;
7 |     default:
8 |         cout << "Undefined key."
9 | }

```

21. How many lines will be printed ?

- A. Compilation error
- B. 10
- C. 11
- D. None of the above

```

1 | for (size_t i = 10; i ≥ 0; i--)
2 |     cout << "Line " << i << endl;

```

22. What's the output of the source code?

- A. 2
- B. 4
- C. 6
- D. 8
- E. The source code cannot be compiled without error.

F. Unpredictable result.

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int idx = 0;
5     int numbers[4] = {2, 4, 6, 8};
6     idx = -1;
7     cout << numbers[idx];
8 }
```

23. What's the output of the source code?

- A. 2
- B. 4
- C. 6
- D. 8
- E. The source code cannot be compiled without error.
- F. Unpredictable result.

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int idx = 0;
5     int numbers[4] = {2, 4, 6, 8};
6     idx = 4;
7     cout << numbers[idx];
8 }
```

24. What's the output if you compile the following source code with C++11 standard?

```
1 char str[16] = {"C++"};
2 cout << strlen(str) << endl;
```

25. What's the output if you compile the following source code with C++11 standard?

```
1 char str {"C++"};
2 cout << str[1] << endl;
```

26. What's the output?

- A. C++
- B. No output.
- C. Compilation error.
- D. Runtime error.

```
1 char str1[16] = {"C++"};
2 char str2[16];
3 str2 = str1;
4 cout << str2;
```

27. What's the output?

- A. C++
- B. No output.
- C. Compilation error.
- D. Runtime error.

```
1 string str1 = {"C++"};
2 string str2;
3 str2 = str1;
4 cout << str2;
```

28. What's the output?

```
1 #include <iostream>
2 union twonumbers {
3     int n[2];
4     double d;
5 };
6 using namespace std;
7 int main () {
8     twonumbers tn;
9     tn.n[0] = 0;
10    tn.n[1] = 0;
11    cout << tn.d;
12    return 0;
13 }
```

29. The output of the following code is:

```
1 struct Person {
2     bool male;
3     int id;
4     char label;
5 }
6 cout << sizeof(struct Person);
```

30. What's the output of the following code?

```
1 int *numbers = new int[8];
2 char *pc = (char *) numbers;
3 *numbers = 0xA0B0C0D;
4 cout << (int) pc[3];
```

31. The following source code is correct.

```
1 int *ptr;
2 *ptr = 3;
```

32. What's the output of the following code on your PC (64 bit OS and CPU)?

```
1 int numbers[8];
2 cout << sizeof(numbers) << endl;
```

33. What's the output of the following code on your PC (64 bit OS and CPU)?

```

1 | int *numbers = new int[8];
2 | cout << sizeof(numbers) << endl;

```

34. The following code can be compiled successfully.

```

1 | double value = 0.0;
2 | const double *p = &value;
3 | value = 2.0;

```

35. The following code can be compiled successfully.

```

1 | double value = 0.0;
2 | double * const p = &value;
3 | p[0] = 2.0;

```

36. The following source code is correct and cannot cause bugs.

```

1 | int *pint = (int *) malloc(8 * sizeof(int));
2 | char * pc = (char *) pint;
3 | pc[8] = 'a';
4 | *(pc + 8) = 'b';

```

37. What's the output of the following code?

```

1 | # include <iostream>
2 | using namespace std;
3 | void foo(float * p) {
4 |     p[0] = 1.0f;
5 | }
6 | int main() {
7 |     float values[4] = {3.0f, 4.0f, 5.0f, 6.0f};
8 |     foo(values + 2);
9 |     cout << *values << " ";
10 |    cout << *values + 2 << " ";
11 |    cout << *(values + 2) << " ";
12 |    cout << values[2] << endl;
13 |    return 0;
14 | }

```

38. What's the output of the following code?

```

1 | # include <iostream>
2 | using namespace std;
3 | struct people {
4 |     string name;
5 |     int age;
6 | };
7 | void init(people p) {
8 |     p.name = "No name";
9 |     p.age = 0;
10 | }
11 | int main() {
12 |     people p;
13 |     p.age = -1;

```

```

14     init(p);
15     cout << p.age << endl;
16     return 0;
17 }

```

39. What's the output of the following code?

- A. 0
- B. -1
- C. < Compile Error >
- D. Random Value

```

1  # include <iostream>
2  using namespace std;
3  struct people {
4      string name;
5      int age;
6  };
7  void init(people * p) {
8      p.name = "No name";
9      p.age = 0;
10 }
11 int main() {
12     people p;
13     p.age = -1;
14     init(p);
15     cout << p.age << endl;
16     return 0;
17 }

```

40. What's the output of the following code?

```

1  # include <iostream>
2  using namespace std;
3  struct people {
4      string name;
5      int age;
6  };
7  void init(people * p) {
8      p → name = "No name";
9      p → age = 0;
10 }
11 int main() {
12     people p;
13     p.age = -1;
14     init(&p);
15     cout << p.age << endl;
16     return 0;
17 }

```

41. What's the output of the following source code?

```

1  # include <iostream>
2  using namespace std;

```

```

3 struct people {
4     string name;
5     int age;
6 };
7 void init(people & p) {
8     p.name = "No name";
9     p.age = 0;
10 }
11 int main() {
12     people p;
13     p.age = -1;
14     init(p);
15     cout << p.age << endl;
16     return 0;
17 }

```

42. What's the output of the following code?

```

1 # include <iostream>
2 using namespace std;
3 float area(float & x) {
4     return x * x;
5 }
6 int main() {
7     float value = 3.0f;
8     cout << area(value);
9 }

```

43. What's the output?

```

1 # include <iostream>
2 using namespace std;
3 int area(int &x);
4 int main() {
5     int n = 10;
6     area(n);
7     cout << n << endl;
8 }
9 int area(int &x) {
10     return x *= x;
11 }

```

44. What's the output of the following code?


```

1  # include <iostream>
2  using namespace std;
3  float area(float & x) {
4      x = x * x;
5      return x;
6  }
7  int main() {
8      float value = 3.0f;
9      cout << area(value) << endl;
10     cout << value << endl;
11     return 0;
12 }

```

45. What's the output of the following code?

- A. 3
- B. 9
- C. < Compile Error >
- D. < None of the above >

```

1  #include <iostream>
2  using namespace std;
3  float area(float * x) {
4      return *x * *x;
5  }
6  int main() {
7      float value = 3.0f;
8      cout << area(value);
9      return 0;
10 }

```

46. What's the output of function `sum()` ?

- A. 4 or 8
- B. 3
- C. 12
- D. 32
- E. < Compile Error >

```

1  # include <iostream>
2  using namespace std;
3  int main() {
4      int cookies[8] = {1, 2, 4, 8, 16, 32, 64, 128};
5      int n = sum_arr(cookies, 3);
6  }
7  int sum_arr(int arr[], int n) {
8      cout << sizeof arr;
9      return 0;
10 }

```

47. What's the output of function `sum()` ?

- A. 4 or 8
- B. 3
- C. 12
- D. 32
- E. < Compile Error >

```

1  # include <iostream>
2  using namespace std;
3  int sum_arr(int arr[], int n) {
4      cout << sizeof arr;
5      return 0;
6  }
7  int main() {
8      int cookies[8] = {1, 2, 4, 8, 16, 32, 64, 128};
9      int n = sum_arr(cookies, 3);
10 }

```

48. There is a function template. The specialization is correctly implemented in the following code.

```

1  template <typename T>
2  T sum(T x, T y) {
3      return x + y;
4  }
5  struct Point {
6      int x;
7      int y;
8  };
9  template Point sum<Point>(Point pt1, Point pt2) {
10     Point pt;
11     pt.x = pt1.x + pt2.x;
12     pt.y = pt1.y + pt2.y;
13     return pt;
14 }

```

49. The following declaration correctly defines some default arguments.

```

1  int harpo(int n = 3, int m, int k = 3);

```

50. The functions and a function pointer are declared as follows. Which answers are correct?

```

1  float norm_l1(float x, float y); //declaration
2  float norm_l2(float x, float y); //declaration
3  float (*norm_ptr)(float x, float y); //norm_ptr is a function pointer

```

- A. `norm_ptr = &norm_l1;`
- B. `norm_ptr = norm_l2;`
- C. `norm_ptr = norm_l2; norm_ptr(3.0f, 4.0f);`
- D. `norm_ptr = &norm_l1; (*norm_ptr)(3.0f, 4.0f);`
- E. `norm_ptr = norm_l1; (*norm_ptr)(3.0f, 4.0f);`

51. The following code correctly defines a function template:

```
1  template <typename T>
2  void swap(T &a, T &b) {
3      T temp = a;
4      a = b;
5      b = temp;
6  }
```

52. Function overloading is that multiple functions share the same function name but different signatures as the two functions below:

```
1  float foo(float arg);
2  int foo(double arg);
```

53. The **this** pointer points to the object and can be used to invoke a member as in the following code.

```
1  class Person {
2      int num;
3      public:
4          static int foo() {
5              return this->num;
6          }
7          // other members
8  };
```

54. What's the output of the following code?

- A. No name
- B. < No output >
- C. < Compilation error >
- D. < Runtime error >

```
1  class Person {
2      string name;
3  };
4  int main() {
5      Person p;
6      p.name = "No name";
7      cout << p.name;
8      return 0;
9  }
```

55. What's the output of the following source code?

```

1  class Hello {
2      static int value;
3      int num;
4      public:
5          int sum(int i, int j) {
6              return i + j;
7          }
8  };
9  int main() {
10     cout << sizeof(Hello) << endl;
11     return 0;
12 }

```

56. What's the output of the source code?

```

1  class Hello {
2      static int value;
3      int num;
4      public:
5          int sum(int i, int j) {
6              return i + j;
7          }
8          void setValue(int v) {
9              value = v;
10         }
11         int getValue() {
12             return value;
13         }
14 };
15 int main() {
16     Hello h1, h2;
17     h1.setValue(5);
18     cout << h2.getValue() << endl;
19     return 0;
20 }

```

- A. 5
- B. A random integer
- C. Compilation error
- D. Link error

57. A class is declared as follows. Please select correct answers for creating a variable.

```

1  class Stock{
2      public:
3          Stock();
4          Stock(const std::string & co, long n = 0, double pr = 0.0);
5          ~Stock();
6          //other members
7  };

```

- A. `Stock st1;`

- B. `Stock st2("MSFT", 3, 2.0f);`
C. `Stock st3 = Stock("MSFT", 3, 2.0f);`

58. Class **Stonewt** is declared as follows.

```
1 class Stonewt {  
2     // some members  
3     public:  
4         Stonewt(double lbs);  
5         Stonewt(int stn, double lbs);  
6         Stonewt();  
7         ~Stonewt();  
8         operator int() const;  
9         operator double() const;  
10 }
```

Which function will be invoked by the following line of code ?

```
1 Stonewt wt = 120;
```

59. Which function will be invoked by the following line of code ?

```
1 wt = 120.0; //wt is an object of type Stonewt
```

60. Which function will be invoked by the following line of code ?

```
1 double f = wt; //wt is an object of type Stonewt
```

【分析】`wt` 是 `Stonewt` 类的一个对象，使用 `operator double() const;` 进行隐式类型转换。

【答案】`operator double() const;`

61. Which function will be invoked by the following line of code ?

```
1 Stonewt wt(120);
```

62. A conversion function is defined outside of the declaration of class **Stonewt** as follow.

```
1 Stonewt::operator double() const {  
2     return pounds;  
3 }
```

63. Assignment operator '=' can be overloaded by a non-member function.

64. We can change operators' precedence by overloading.

65. If the friend function is defined as in the following source code, it is a member function in the class.

```
1 class Time {  
2     private:  
3         int hours;  
4         int minutes;  
5     public:  
6         Time();  
7         Time(int h, int m = 0);
```

```

8      void AddMin(int m);
9      void AddHr(int h);
10     void Reset(int h = 0, int m = 0);
11     Time operator + (const Time & t) const;
12     Time operator - (const Time & t) const;
13     Time operator * (double n) const;
14     void show() const;
15     friend Time operator * (double mult, Time in);
16 };

```

66. If we define a member function as follows for class Time

```

1 | Time Time::operator*(double mult) const

```

then we can calculate as follows

```

1 | a = 3.3 * b; //a and b are objects of type Time

```

67. operator+() overloads the + operator, and it can only be used for mathematical addition.

68. You can define two constructors as follows for class Person.

```

1 | Person(){...}
2 | Person(int m = 0) {...}

```

69. Please read the following code and choose correct answers:

```

1 | class Person {
2 |     char *name;
3 |     public:
4 |         Person() {
5 |             name = new char [128];
6 |         }
7 |         ~Person() {
8 |             delete name;
9 |         }
10 | };

```

- A. The code can be compiled without error.
- B. Runtime error.
- C. It can cause memory double free problem.
- D. It can cause memory leak.

70. For class Person, which of the constructors is a copy constructor?

- A. `Person::Person();`
- B. `Person::Person(const Person & p);`
- C. `Person::Person(int m);`
- D. `Person::Person(int m, int n);`

71. For class Person, which of the constructors is its default constructor?

- A. `Person::Person();`
- B. `Person::Person(const Person & p);`
- C. `Person::Person(int m);`

D. `Person::Person(int m, int n);`

72. If assignment operator is not defined in class Person, the following code will invoke default assignment operator.

```
1 | p1 = p2 = p3; //p1, p2 and p3 are objects of type Person
```

73. If you do not define a default constructor for a class explicitly, then no default constructor for that class.

74. What is the output?

```
1 | class Animal {
2 |     private:
3 |         int weight;
4 |     public:
5 |         Animal(int w = 0) {
6 |             weight = w;
7 |         }
8 |         void print() {
9 |             cout << weight << endl;
10 |        }
11 | };
12 | class Dog: public Animal {
13 |     public:
14 |         Dog(int w = 0): Animal(w) {}
15 |         void print() {
16 |             cout << "Dog ";
17 |             Animal::print();
18 |         }
19 |         void speak() {
20 |             cout << "wangwang" << endl;
21 |         }
22 | };
23 | int main() {
24 |     Dog dog(5);
25 |     Animal * p = & dog;
26 |     p -> print();
27 |     return 0;
28 | }
```

75. What is the output?

A. wangwang

B. < Compilation Error >

C. None of above

```
1 | class Animal {
2 |     private:
3 |         int weight;
4 |     public:
5 |         Animal(int w = 0) {
6 |             weight = w;
7 |         }
8 |         void print() {
9 |             cout << weight << endl;
```

```

10     }
11 };
12 class Dog: public Animal {
13     public:
14         Dog(int w = 0): Animal(w) {}
15         void print() {
16             cout << "Dog ";
17             Animal::print();
18         }
19         virtual void speak() {
20             cout << "wangwang" << endl;
21         }
22 };
23 int main() {
24     Dog dog(5);
25     Animal * p = & dog;
26     p → speak();
27     return 0;
28 }

```

76. What is the output?

```

1  class Animal {
2      private:
3          int weight;
4      public:
5          Animal(int w = 0) {
6              weight = w;
7          }
8          virtual void print() {
9              cout << weight << endl;
10         }
11 };
12 class Dog: public Animal {
13     public:
14         Dog(int w = 0): Animal(w) {}
15         void print() {
16             cout << "Dog ";
17             Animal::print();
18         }
19         void speak() {
20             cout << "wangwang" << endl;
21         }
22 };
23 int main() {
24     Dog dog(5);
25     Animal * p = & dog;
26     p → print();
27     return 0;
28 }

```

77. Please choose the right answer(s) for declaring a class template

- A. `template <class Type> class ClassName{...}`
- B. `template <typename Type> class ClassName{...}`

78. Matx and Matx12f are declared in the following figure. Please choose the correct statement(s).

- A. Matx is a class tempate.
- B. Matx is a tempate class.
- C. Matx12f is a class template.
- D. Matx12f is a template class.

```
1  template <typename _Tp, int m, int n> class Matx {
2      public:
3          enum {
4              rows = w,
5              cols = n,
6              channels = rows * cols,
7  #ifdef OPENCV_TRAITS_ENABLE_DEPRECATED
8              depth = traits::Type<_Tp>:: value,
9              type = CV_MAKETYPE(depth, channels),
10 #endif
11              shortdim = (m < n ? m : n)
12          };
13 };
14 typedef Matx<float, 1, 2> Matx12f;
15 typedef Matx<double, 1, 2> Matx12d;
16 typedef Matx<float, 1, 3> Matx13f;
17 typedef Matx<double, 1, 3> Matx13d;
18 typedef Matx<float, 1, 4> Matx14f;
19 typedef Matx<double, 1, 4> Matx14d;
20 typedef Matx<float, 1, 6> Matx16f;
21 typedef Matx<double, 1, 6> Matx16d;
```

79. What's the output of the following code?

```
1  #include <iostream>
2  using namespace std;
3  template <typename Type>
4  class Stack {
5      private:
6          Type items[16];
7      public:
8          size_t size() {
9              return sizeof(items);
10         }
11 };
12 int main() {
13     Stack<float> st;
14     cout << st.size() << endl;
15     return 0;
16 }
```

- 80. When an exception is thrown, the program must be terminated.
- 81. A try block can be followed by multiple catch blocks.
- 82. The following source code cannot be compiled successfully.

```

1 double gmean(double a, double b) {
2     if (a < 0 || b < 0)
3         throw string("bad arguments");
4     return std::sqrt(a * b);
5 }
6 int main() {
7     try {
8         gmean(3, -3);
9     }
10 }

```

83. The following code cannot be compiled successfully since 'try' is commented.

```

1 double gmean(double a, double b) {
2     if (a < 0 || b < 0)
3         throw string("bad arguments");
4     return std::sqrt(a * b);
5 }
6 int main() {
7     gmean(3, -3);
8 }

```