

SECTION A True or False

Use the answer sheet provided for answering the questions in this section. Each correct answer will garner 1 mark.

1. ~~An abstract class is a class that has at most one pure virtual function member.~~

~~(a) True~~

~~(b) False~~

Answer: (b) False

2. ~~One key difference between C++ structures and C structures is that C++ structures can have member functions.~~

~~(a) True~~

~~(b) False~~

Answer: (a) True

3. In the assembly phase, the compiler translates a pre-processed C/C++ source code into an assembly file.

(a) True

(b) False

Answer: (b) False

4. In C and C++, there is no check on whether an array index is out of bounds. If an array index goes out of bounds, the program *always* terminates in an error.

(a) True

(b) False

Answer: (b) False

5. Arrays can be initialised during their declaration. If there are fewer initial values than the array size, the remaining elements are initialized to 0.

(a) True

(b) False

Answer: (a) True

6. The base address of an array is the address of the first array component.

(a) True

(b) False

Answer: (a) True

7. The following C code will compile without errors:

```
int foo(const int * a, const int * b)
{
    *a = 5;
    return *a + *b;
}
```

(a) True

(b) False

Answer: (b) False

~~8. In C++, the statement `delete p;` deallocates the variable pointer `p`.~~

~~(a) True~~

~~(b) False~~

~~Answer: (b) False~~

SECTION B Multiple Choice

Use the answer sheet provided for answering the questions in this section.

Each correct answer will garner 1 mark.

9. Consider the following statement:

```
char str[] = "Twelve";
```

What is the size of the array `str`?

- (a) The statement will cause a syntax error.
- (b) 6
- (c) 7
- (d) None of the above

Answer: (c)

10. Consider the following code fragment:

```
int i = 4, j = 0;
while(--i) { j++; }
```

What is the value of the variable `j` after the completion of the while-loop?

- (a) The code will not compile because of syntax error in the while-loop condition.
- (b) 2
- (c) 3
- (d) 4

Answer: (c)

11. Consider the following code snippet:

```
union {
    char c;
    short s;
    int i;
} u;
```

```
u.c = 'A';
```

What is the size of the variable `u` be equal to?

- (a) `sizeof(char)`
- (b) `sizeof(short)`
- (c) `sizeof(int)`
- (d) None of the above

Answer: (c)

12. Consider the following function-like macro:

```
#define FLM(X,Y) X/Y
```

To what value does the macro evaluate to when invoked as `FLM(2 + 9, 3 - 2)`?

- (a) 11
- (b) 3
- (c) 1
- (d) None of the above

Answer: (b)

13. Consider the following C/C++ code snippet:

```
enum loudness { faint = -1, moderate, defeaning = 2, painful };
```

What is the value of `moderate`?

- (a) 0
- (b) 1
- (c) 2
- (d) None of the above

Answer: (a)

14. In C and C++, generic pointers can be declared with

- (a) `static`
- (b) `void`
- (c) `const`
- (d) None of the above

Answer: (b)

15. Given the declaration below:

```
char name[30];
```

Which of the following statements are *invalid*?

- i. `name = "Terakihi";`
- ii. `strcpy(name, "Snapper");`
- iii. `name = {'H', 'a', 'k', 'e'};`
- iv. `name[0] = 'G';`

- (a) i and iii
- (b) ii and iv
- (c) i and ii
- (d) They are all valid

Answer: (a)

16. ~~What is the output from the following C++ code?~~

```
#include <iostream>
using namespace std;
int main(void)
{
    int * ptr = new int;
    cout<<ptr<<" - "<<*ptr;
    return 0;
}
```

~~(a) (Address of memory allocated) (Garbage value)~~

~~(b) (Address of memory allocated) 0~~

~~(c) (Address of ptr) 0~~

~~(d) (Address of ptr) (Garbage value)~~

Answer: (a)

17. ~~Given the C++ declaration:~~

```
char name[8] = "Marlin";
```

~~Which of the following statements output Marlin?~~

~~i. std::cout << name;~~

~~ii. for(int j=0; j<6; j++) std::cout << name[j];~~

~~iii. int j=0; while (name[j] != '\0') std::cout << name[j++];~~

~~iv. int j=0; while (j < 8) std::cout << name[j+1];~~

~~(a) All of the above~~

~~(b) None of the above~~

~~(c) i, ii and iii~~

~~(d) i, iii and iv~~

Answer: (c)

18. What is the name of the & operator in relation to pointers?

(a) Conditional operator

(b) Logical operator

(c) Address of operator

(d) None of the above

Answer: (c)

19. What is the output of the following program?

```
#include<stdio.h>
int main()
{
    int a[5] = {2, 3};
    printf("%d,%d,%d\n", a[2], a[3], a[4]);
    return 0;
}
```

(a) Garbage values

(b) 2,3,3

(c) 3,2,2

(d) 0,0,0

Answer: (d)

20. ~~What would be the equivalent pointer expression for referring the array element $a[i][j][k][l]$?~~

~~(a) $((((a+i)+j)+k)+l)$~~

~~(b) $*(*(*(*(a+i)+j)+k)+l)$~~

~~(c) $((a+i)+j)+k+l)$~~

~~(d) $((a+i)+j+k+l)$~~

Answer: (b)

SECTION C Short Answer**Write your answer in the space provided.**

21. What are the four phases of compilation in C and C++?
- (8 marks)**

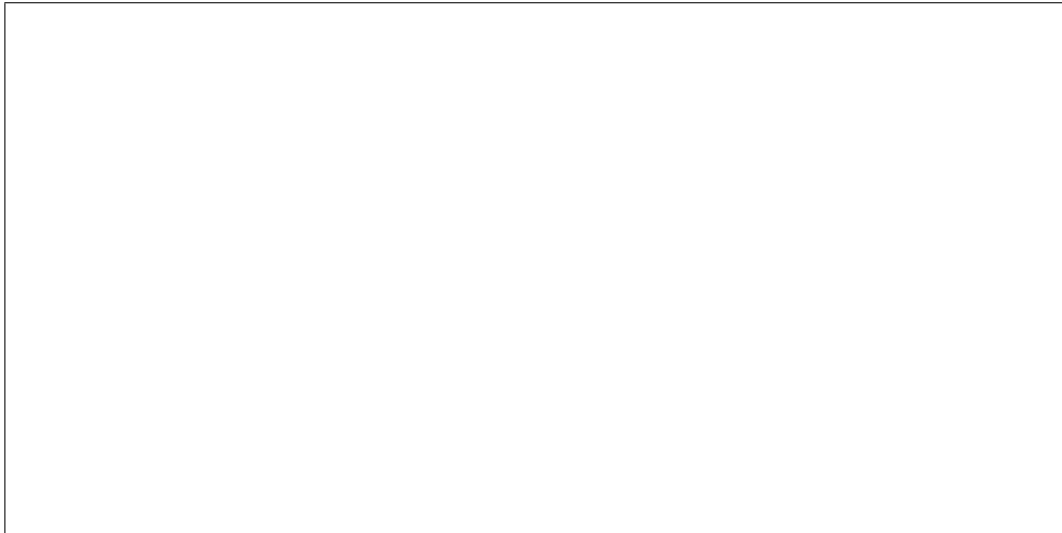
Pre-processing
Assembly
Compilation
Linking

22. Consider the following C++ class declaration:
- (5 marks)**

```
namespace nsA {  
    class ClassA {  
    public:  
        virtual int f1() const = 0;  
        virtual void f2() = 0;  
    protected:  
        int a;  
    };  
}
```

~~Declare a class ClassB that extends ClassA but in a different namespace called nsB. ClassB should preserve the access specifier of the members, should not be abstract, and should have an inline default constructor that initializes the member variable a to 100.~~

~~(Hint: You do not need to show function implementations, just the prototype declarations)~~

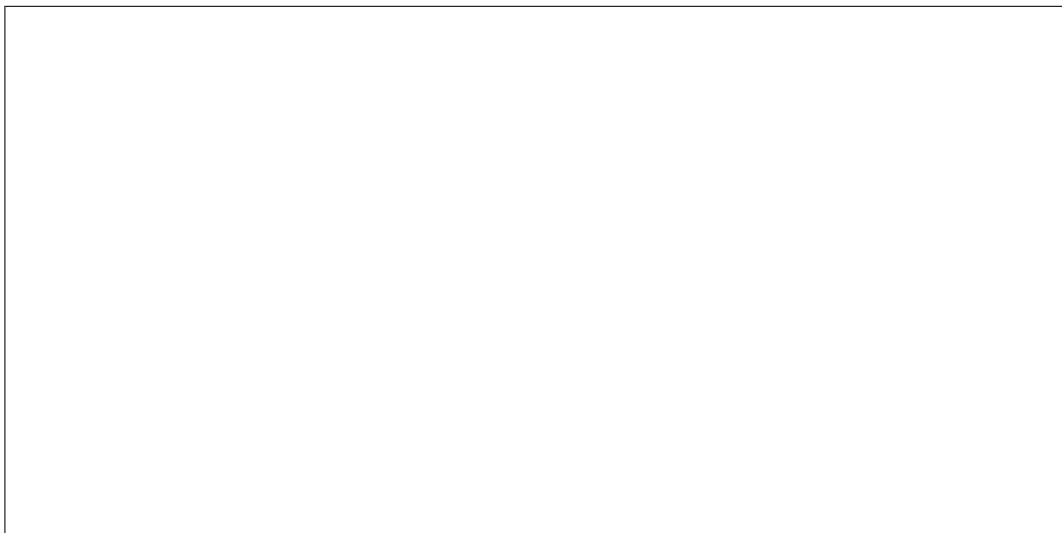


```
namespace nsB {  
    class ClassB: public nsA::ClassA {  
    public:  
        int f1() const;  
        void f2();  
        ClassB() : a(100) {}  
    };  
}
```

23. Given these two variable declarations in a C program:

```
char str1[] = "I am a string.\n";  
char *str2 = "I am a string.\n";
```

(a) What is the difference between the two statements? **(1 mark)**



str1 is an array of characters that contains a string variable. str2 is a pointer to a string literal or constant.

Or

The first string is stored as a string variable while the second string is stored as a string literal or constant.

(b) Write a statement to output the letter "s" in `str1` using `printf`. Use array index to refer to the element. **(2 marks)**

```
printf("%c",str1[7]);
```

(c) Write a statement to output the letter "s" in `str2` using `printf`. Use appropriate pointer arithmetic and operator to refer to the element. **(2 marks)**

```
printf("%c",*(str2+7));
```

(d) Write some code to copy the second string to the first. Which header file do you need to include? **(2 marks)**

(Hint: The function `char *strcpy(char *dest, const char *src)` copies the string pointed to by `src` to `dest`.)

Code to copy second string to first:

```
strcpy(str1, str2);
```

Header file to include:

```
string.h
```

24. Given the following code:

```
char Z, Y;  
Z = 'g';  
Y = '#';  
char * pZ,* pY;  
pY = &Y;  
pZ = &Z;  
pY = pZ;
```

Suppose each char occupies 1 byte of memory and the variable Z is at (decimal) address 2304 and the variable Y is at (decimal) address 2305. After the code above has run:

(a) What value is represented by &Y? (1 mark)

2305

(b) What value is represented by pZ? (1 mark)

2304

(c) What value is represented by pY? (1 mark)

2304

(d) What value is represented by *pZ? (1 mark)

'g'

(e) What is the value represented by *pY (1 mark)

'g'
