



*Department of Computer Science & Software Engineering  
The University of Western Australia*

**Mid-Semester Test  
September 2020**

**Systems Programming 2002 (CITS2002)**

This paper contains: 8 pages (including this title page)  
Time allowed: 45 minutes

Each of the 20 questions is worth 1 mark

You should attempt ALL questions

**PLEASE NOTE**

*This test is formative assessment, and does not contribute to the unit's overall marks.*

*This test is optional, and you do not need to submit your answers anywhere. Solutions will be posted online, and you will be able to mark your own work.*

*While this is a 'take home test', you are encouraged to complete it as if it were a 'closed book test' – within the time limit, and without reference to any written or online materials.*

1.

- (1) Which of the following statements best describes a compiler that conforms to the C99 standard?
  - A. The compiler will compile only programs employing C99-specific features.
  - B. The compiler will detect all errors within a C99 program.
  - C. The compiler will compile all correct C99 programs, regardless of the operating system on which it is running.
  - D. The compiler will generate portable machine-code, executable across all different machine architectures.
  
- (2) Which of the following words is a valid C preprocessor directive?
  - A. `#import`
  - B. `#defined`
  - C. `#undefined`
  - D. `#endif`
  
- (3) Which of the following words is a valid C99 keyword?
  - A. `function`
  - B. `class`
  - C. `do`
  - D. `array`
  
- (4) Assuming a program is always supplied with at least one command-line argument, which of the following expressions returns the last command-line argument supplied?
  - A. `argv[0]`
  - B. `argv[1]`
  - C. `argv[argc-1]`
  - D. `argv[argc]`

(5) Consider the following C99 function:

```
void go(int x)
{
    int x = 5;
    {
        int x = 6;
    }
    printf("x = %i\n", x);
}
```

Which best describes what happens when we attempt to compile and run this code?

- A. A compile-time error is generated because the variable `x` is multiply defined.
- B. The value '5' is printed.
- C. The value '6' is printed.
- D. A run-time error is generated because C doesn't know which value of `x` to print.

(6) Consider the following function:

```
void go(void)
{
    for (int i = 10; i > 0; --i)
    {
        if ((i % 3) == 1)
        {
            continue;
        }
        printf("%i  ", i);
    }
    printf("\n");
}
```

What is printed when the `go` function is executed?

- A. 10 8 7 5 4 2 1
- B. 10 8 7 5 4 2 1 0
- C. 9 8 6 5 3 2
- D. 9 6 3 0

(7) Consider the following function:

```
void go(void)
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = i; j < 5; j++)
        {
            printf("*");
        }
        printf("!\n");
    }
}
```

What is printed when the go function is executed?

- A. !  
\*!  
\*\*!
- B. \*!  
\*\*!  
\*\*\*!
- C. \*\*\*\*\*!  
\*\*\*\*!  
\*\*\*!
- D. \*\*\*!  
\*\*\*\*!  
\*\*\*\*\*!

(8) Consider a function of type void. Which of the following statements is true?

- A. The function must have precisely 0 return statements.
- B. The function must have precisely 1 return statement.
- C. The function must have at least 1 return statement.
- D. The function may have any number of return statements.

(9) Consider the following array declaration:

```
char string[20];
```

Which of the following code fragments moves all the characters in index positions 0 through 4 inclusive one place to the right to index positions 1 through 5 inclusive?

- A. 

```
for (int i = 5; i > 0; i--)  
{  
    string[i] = string[i-1];  
}
```
- B. 

```
for (int i = 1; i <= 5; i++)  
{  
    string[i] = string[i+1];  
}
```
- C. 

```
for (int i = 0; i < 5; i++)  
{  
    string[i+1] = string[i];  
}
```
- D. 

```
for (int i = 4; i >= 0; i--)  
{  
    string[i] = string[i+1];  
}
```

(10) Which of the following statements about how lines are terminated in text files is true?

- A. The end of a line is represented by the null-byte character.
- B. The end of a line is represented by the newline character followed by the null-byte character.
- C. The end of a line is represented by the carriage-return character followed by the end-of-line character.
- D. The way the end of a line is represented is operating-system dependent.

(11) Consider the following C99 variable definition:

```
int buffer[100];
```

What can we say about the expression `sizeof buffer` ?

- A. The expression has the value 400.
- B. As we do not know what is stored in `buffer`, we cannot know the value of the expression until runtime.
- C. The value of the expression is defined to be undefined in pre-C99 programs, but defined to be 800 in C99 programs.
- D. The value of the expression is operating system and compiler dependent.

- (12) Which one of the following is not the responsibility of an operating system kernel:
- A. managing the resources in a computer.
  - B. allocating memory for user processes.
  - C. repairing the electronic circuits in a computer.
  - D. providing an interface for user processes to access input/output devices.
- (13) Which one of the following best characterises a process:
- A. a process is a set of instructions followed by a software engineer to develop a program.
  - B. a process is a utility program provided by an operating system.
  - C. a process is the methodology for learning to write programs.
  - D. a process is a computer program undergoing execution.
- (14) Operating systems started supporting multiprogramming because:
- A. programmers started writing multiple programs.
  - B. it was the best way of keeping the processor busy.
  - C. AI programs started writing other programs.
  - D. as RAM capacity grew, computers could store multiple programs.
- (15) What happens if the `fopen` function fails to open a file for writing?
- A. the process is terminated by the operating system.
  - B. the system changes the file mode to writable and attempts to re-open the file.
  - C. execution continues normally.
  - D. an error message is printed to the `stderr` stream.
- (16) A “blocked process” is a process that:
- A. has made an input/output request.
  - B. has requested an undefined operation, that cannot be performed.
  - C. has been blocked by the operating system because it executed an illegal instruction.
  - D. tries to block the user from using a computer.

- (17) A process undergoes a state transition from “blocked” to “blocked suspend” when:
- A. it executes an illegal instruction.
  - B. the operating system frees up memory to accommodate other processes.
  - C. the operating system determines that the process has exceeded its time quantum.
  - D. the operating system detects that the process is a computer virus.
- (18) What is the best explanation for why an operating system may manage blocked processes in multiple queues?
- A. to simplify searching by having a different queue for each user.
  - B. to simplify searching by having a different queue for each device.
  - C. to simplify searching by having a different queue for each process.
  - D. to support non-deterministic scheduling by resuming a process from a random-access queue.
- (19) Consider the following C code:

```
int main(int argc, char *argv[])
{
    if(fork() != getpid())
        printf("I'm alive!\n");
    return 0;
}
```

The message: "I'm alive!" will be printed by:

- A. the child process.
  - B. the parent process.
  - C. both the child and the parent process.
  - D. neither the child nor the parent process.
- (20) Why do C libraries and the operating system kernel both check the validity of system call parameters?
- A. user-written checks are more likely to be flawed than those of the kernel.
  - B. it is possible to bypass the libraries' checks and call the system calls directly.
  - C. in practice, the kernel doesn't need to check system call parameters if the library's checks are rigorous.
  - D. checking parameters twice ensures that they haven't changed between checks.
-

## Solutions

1.

1. C
  2. D
  3. C
  4. C
  5. A
  6. C
  7. C
  8. D
  9. A
  10. D
  11. D
  12. C
  13. D
  14. B
  15. C
  16. A
  17. B
  18. B
  19. C
  20. B
-