Name: CS520

Midterm Exam

16 October 2018, 80 minutes, 21 questions, 100 points

The exam is closed book and notes.
Please keep all electronic devices turned off and out of reach.
Note that a question may require *multiple* checked boxes for a correct answer. Checking *some* but not *all* of the required boxes will result in a *partial* answer worth only 2 of the 5 points. Checking any box that shouldn't be checked results in an *incorrect* answer, worth zero.

1. ◯ Hold my exam in your office. I will pick it up prior to December 11. [0 pts]
   ◯ Shred my exam. I never want to see it again.
   √ Return my exam to my Kingsbury mailbox.

2. How would the following C string (i.e. null terminated) `"xyz"` be represented using ASCII in the memory of a Little Endian machine with a byte-addressable memory? The bytes are shown below in increasing memory address order, left to right. The ASCII code for `'x'` is `0x78`. The ASCII code for `'y'` is `0x79`. The ASCII code for `'z'` is `0x7A`. [5 pts]
   ◯ `0x00 0x7A 0x79 0x78`.
   ◯ `0x78 0x79 0x7A`.
   ◯ `0x7A 0x79 0x78`.
   √ `0x78 0x79 0x7A 0x00`.
   ◯ none of the above.

3. Interpret `0xD0` and `0xD1` as 8-bit two's complement integers and add them together to produce an 8-bit two's complement integer. The result is: [5 pts]
   ◯ `0xA2`.
   ◯ `0xDF`.
   √ `0xA1`.
   ◯ `0x80`.
   ◯ none of the above.

4. Consider the following C function: [5 pts]

```
signed int f(void)
{
  int i = 1;
  return *(signed char *) &i;
}
```

   On a machine with a byte-addressable memory, and with 32-bit integers, the function will:
   ◯ always return 0.
   ◯ always return 1.
   ◯ return 1 if the machine is big-endian and 0 otherwise.
   √ return 1 if the machine is little-endian and 0 otherwise.
   ◯ none of the above.

5. Which of the following 16-bit two's complement values will overflow when they are moved to an 8-bit two's complement container? [5 pts]

   ◯ 0xFFFF.

   √ 0x8000.

   √ 0x0087.

   √ 0x908A.

6. What would -73 look like as a 32-bit two's complement integer in the memory of big-endian machine? The bytes are shown below in increasing memory address order, left to right. [5 pts]

   ◯ 0xB6 0xFF 0xFF 0xFF.

   ◯ 0xB5 0xFF 0xFF 0xFF.

   ◯ 0x80 0x00 0x00 0x49.

   ◯ 0xFF 0xFF 0xFF 0xB6.

   ◯ 0x49 0x00 0x00 0x80.

   √ none of the above.

7. Which of the following statements about IEEE single-precision floating-point are true? [5 pts]

   ◯ It stores the exponent as a two's complement value.

   √ There are two infinity values, positive infinity and negative infinity.

   √ Denormalized values have a stored exponent with all 0 bits and a stored significand that is not all 0 bits.

   √ NaN values have a stored exponent with all 1 bits and a stored significand that is not all 0 bits.

8. Represent 73.5 as an IEEE single-precision floating-point value. Its bits in hex would be: [5 pts]

   ◯ 03130000.

   √ 42930000.

   ◯ 42C98000.

   ◯ C2C98000.

   ◯ none of the above.

9. Interpret 0x00111111 and 0x00222222 as IEEE single-precision floating-point values. Which of the following statements are true? [5 pts]

   ◯ 0x00111111 is greater than 0x00222222

   √ They are both positive values.

   √ They are both denormalized values.

   ◯ They are both NaN values.

10. Interpret 0x42A0000111111111 as IEEE double-precision floating-point and convert it to IEEE single-precision. What is the result in hex? [5 pts]

   ◯ 0x55100009.

   ◯ 0x55100008.

   √ 0x55000009.

   ◯ 0x55000008.

   ◯ none of the above.

11. Interpret `0x00C44444` as an IEEE single-precision floating point value. Convert it to IEEE double-precision floating point. What is the result in hex? [5 pts]

√ `0x3818888880000000`.

◯ `0x381C444440000000`.

◯ `0x47D8888800000000`.

◯ `0x47DC444440000000`.

◯ none of the above.

12. Represent this UTF-16 value, `0x1234`, in UTF-8. What is the result? [5 pts]

◯ `0x12 0x34`.

◯ `0xE1 0x8C 0xB4`.

√ `0xE1 0x88 0xB4`.

◯ `0xF0 0x81 0x88 0xB4`.

◯ none of the above.

13. Represent this UTF-32 value, `0x00030303`, in UTF-16. What is the result? [5 pts]

◯ `0x0003 0x0303`.

√ `0xD880 0xDF03`

◯ `0xDF03 0xD880`.

◯ `0xDF0C 0xD880`.

◯ none of the above.

14. Which of the following statements are true? [5 pts]

√ The UTF-16 encoding of an ASCII character is always two bytes long.

◯ The UTF-8 encoding of an ASCII character is always two bytes long.

◯ The UTF-8 encoding of a Unicode character is always longer than the UTF-16 encoding of the character.

◯ The UTF-8 encoding of a Unicode character is always shorter than the UTF-32 encoding of the character.

15. Which of the following UTF-8 sequences contain at least one error? [5 pts]

√ `0xF0 0x80 0x80 0x81`.

◯ `0xF1 0x80 0x80 0x81`.

◯ `0xCF 0x81`.

√ `0x91`.

16. Which of the following statements about a class file are true? [5 pts]

√ The code for a method is stored within the Code attribute for the method.

◯ All class files are the same length.

√ All class files use Big Endian format.

◯ All constant pool entries are the same length.

17. Consider the following assembly code fragment for the Java Virtual Machine: [5 pts]

```
top:
  iconst_1
  iload_1
  if_icmpeq top
```

What is the encoding of the two-byte address (offset) stored in the `if_icmpeq`) instruction? (The `iconst_1` and `iload_1` instructions are one-byte instructions. The `if_icmpeq` instruction is a three-byte instruction.)

- ◯ `0x8002`.
- ◯ `0x8005`.
- √ `0xFFFE`.
- ◯ `0xFFFB`.
- ◯ none of the above.

18. Which of the following statements about linking are true? [5 pts]

- √ The linker matches up the definition of a symbol in one file with references to the symbol in the other file.
- ◯ The linker combines input assembly language files into a single output assembly language file.
- √ If a symbol is defined in both input files to a linker, then it is duplicate symbol error.
- ◯ If a symbol is referenced in both input files to a linker, but is not defined in either file, then it is duplicate symbol error.

19. Which of the following statements about the Java Virtual Machine (JVM) are true? [5 pts]

- √ All instructions start with a one-byte opcode.
- ◯ The JVM has registers used for storing the operands and the results of arithmetic instructions.
- √ The JVM has a register, the program counter, which contains the address of the instruction currently being executed.
- ◯ All JVM instructions are the same length.

20. Which of the following statements about PC-relative addresses are true? [5 pts]

- √ The Java Virtual Machine uses PC-relative addresses in its branch instructions.
- √ At execution time they are added to the program counter to get the actual address.
- √ They can contain both negative and positive values.
- ◯ The PC in PC-relative stands for "politically correct."

21. Which of the following statements are true? [5 pts]

- ◯ An assembler has two passes because the definition for a label may come before all the references to the label.
- ◯ The first pass of an assembler only exists to determine the length of the input file.
- √ An assembler reads an assembly language source file as input and writes an object file as output.
- √ The second pass of an assembler determines the address of a referenced label by looking it up in a symbol table created by the first pass of the assembler.