CSE232 US22 Mid-Term Exam (Practice)

This is a practice exam for the CSE232 mid-term exam. Note: these questions are similar in nature but not identical to the actual mid-term exam.

1. **Given the declaration int i = 5; what does i++; return?**
   (a) 0
   (b) 5
   (c) 6
   (d) Undefined
   (e) None of the above

2. **For the following code snippet:**
   
   ```
   int val1 = 10;
   int& val2 = val1;
   ```

   **What does the & signify?**
   (a) Take the Boolean-and of val1 and val2
   (b) Treat val2 like a reference type
   (c) Extract the memory address of val2
   (d) This expression is illegal (it will not compile)
   (e) None of the above

3. **For the following code snippet:**
   
   ```
   int val1 = 123;
   int* val2 = &val1;
   ```

   **What does the & signify?**
   (a) Take the Boolean-and of val1
   (b) Treat val1 like a reference type
   (c) Extract the memory address of val1
   (d) This expression is illegal (it will not compile)
   (e) None of the above
4. **Given the decimal value 17, which of the following statements are true?**
   (a) Hexadecimal equivalent is 0x11
   (b) Binary equivalent is 0b10001 (C++14 notation)
   (c) Octal equivalent is 021
   (d) All of the above
   (e) None of the above

5. **What is the return type of the expression `std::cout << 14;`?**
   (a) No return, a void type
   (b) long
   (c) int
   (d) char
   (e) None of the above

6. **Which of the following are true about the terms *lvalue* and *rvalue*?**
   (a) An rvalue represents a value
   (b) An lvalue represents a memory location
   (c) rvalue is on the right and lvalue is on the left of an assignment statement
   (d) All of the above
   (e) None of the above
Note: Refer to the following code snippet for questions 7 – 12.

```cpp
#include <iostream>

int main() {
    int val1 = 2;
    double val2 = 3.4;
    int& val1_ref = val1;
    int* val1_ptr = &val1;

    std::cout << std::boolalpha;
    std::cout << val1_ref << std::endl;             // Line 1
    std::cout << (&val1_ref == &val1) << std::endl; // Line 2

    val1 = 4;
    std::cout << val1_ptr << std::endl;             // Line 3

    auto result = *val1_ptr * val2;                 // Line 4
    std::cout << result << std::endl;               // Line 5
    std::cout << val1_ref++ << std::endl;           // Line 6
}
```

7. **What is the output produced by Line 1?**
   (a) 0
   (b) 1
   (c) 2
   (d) 4
   (e) None of the above

8. **What is the output produced by Line 2?**
   (a) true
   (b) false
   (c) 0
   (d) 1
   (e) None of the above
9. **What is the output produced by Line 3?**
   (a) some address
   (b) 1
   (c) 2
   (d) 4
   (e) None of the above

10. **What is the type of result on Line 4?**
    (a) int
    (b) int reference
    (c) int pointer
    (d) bool
    (e) None of the above

11. **What is the output produced by Line 5?**
    (a) 3.4
    (b) 13.6
    (c) 6.8
    (d) 2.0
    (e) None of the above

12. **What output is produced by Line 6?**
    (a) some address
    (b) 1
    (c) 2
    (d) 4
    (e) None of the above
Note: Refer to the following code snippet for questions 13 - 14.

```cpp
std::string fn1(std::string text, int length) {
    for (int i = 0; i < length; ++i) {
        text = text.substr(1) + text[0];
    }
    return text;
}
```

13. If s = "abcdef", what is the result of calling fn1(s, 2)?

14. If s = "abcdef", What is the value of s after calling fn1(s, 2)?

Note: Refer to the following code snippet for questions 15 - 17.

```cpp
std::string fn2(std::string text, char c1, char c2) {
    std::string result;
    for (auto element : text) {
        if ((element < c1) || (element > c2)) {
            result += element;
        }
    }
    return result;
}
```

15. What is the type of the element variable in fn2?

16. What is the result of calling fn2("abcdefg", 'c', 'e')?

17. What is the result of calling fn2("abcdefg", 'e', 'c')?

Note: Refer to the following code snippet for questions 18 - 19.

```cpp
int fn3(double val1, double val2) {
    int vall_int = static_cast<long>(val1);
    double vall_rem = val1 - vall_int;
    if (vall_rem >= val2) {
        return vall_int + 1;
    } else {
        return vall;
    }
}
```
18. What is the result of calling fn3(123.75, 0.5)?

19. What is the result of calling fn3(17.5, 0.5)?

Note: Refer to the following code snippet for questions 20 - 21.

```c
int fn4(int val) {
    int result = 0;
    for (int i = 1; i <= val; ++i) {
        if (val % i == 0) {
            result += i;
        }
    }
    return result;
}
```

20. What is the result of calling fn4(12)?

21. What is the result of calling fn4(31)?

Note: Refer to the following code snippet for questions 22 - 23.

```c
int fn5(int val) {
    int result = 0;
    int rem = 0;
    while (val > 0) {
        rem = val % 10;
        result += rem;
        val /= 10;
    }
    return result;
}
```

22. What is the result of calling fn5(12)?

23. What is the result of calling fn5(31)?
Note: Refer to the following code snippet for questions 24 - 25.

```c
int fn6(int val) {
    return (val % 2) ? val * 2 : val / 2;
}
```

24. What is the result of calling `fn6(12)`?
25. What is the result of calling `fn6(31)`?

Note: Refer to the following code snippet for questions 26 - 27.

```c
int fn7(int n) {
    int result = 0;
    for (int i = 1; i < n; i += 3) {
        if (i % 2) {
            result *= 2;
        } else {
            result += i;
        }
    }
    return result;
}
```

26. What is the result of calling `fn7(9)`?
27. How many iterations of the for loop occur when calling `fn7(9)`?