

# Sample Exam Week 12

## CSE 232 (Introduction to Programming II)

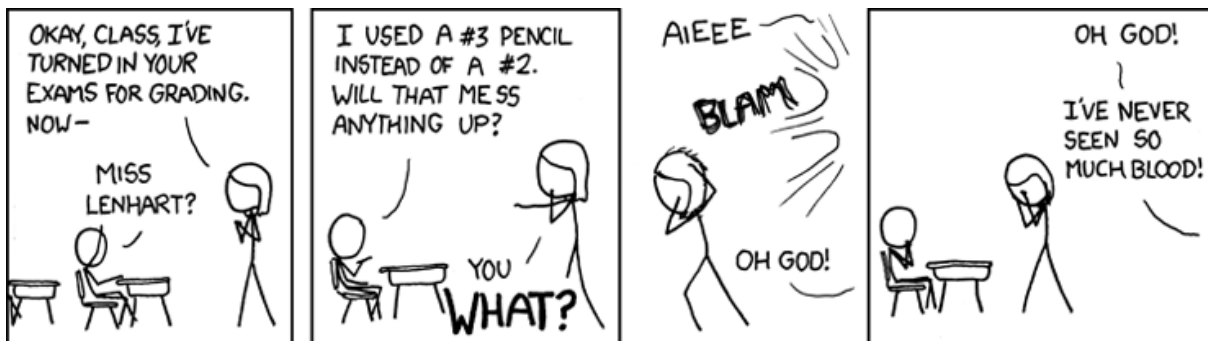
### VERSION A

Full Name: .....

Student Number: .....

#### Instructions:

- DO NOT START/OPEN THE EXAM UNTIL TOLD TO DO SO.
- You may however write and bubble in your name, student number and exam **VERSION/FORM NUMBER** (with a #2 pencil) on the front of the printed exam and bubble sheet prior to the exam start. This exam is Version A. Your section doesn't matter and can be ignored.
- Present your MSU ID (or other photo ID) when returning your bubble sheet and printed exam.
- Only choose one option for each question. Please mark the chosen option in both this printed exam and the bubble sheet.
- Assume any needed `#includes` and `using std::...;` namespace declarations are performed for the code samples.
- Every question is worth the same amount of points. There are 55 questions, but you only need 50 questions correct for a perfect score.
- No electronics are allowed to be used or worn during the exam. This means smart-watches, phones and headphones need to be placed away in your bag.
- The exam is open note, meaning that any paper material (notes, slides, prior exams, assignments, books, etc.) are all allowed. Please place all such material on your desk prior to the start of the exam, (so you won't need to rummage in your bag during the exam).
- If you have any questions during the exam or finish the exam early, please raise your hand and a proctor will attend you.



<http://xkcd.com/499/>

1. Consider the function below. Which standard library function would do the same job with the same arguments?

```
template <typename ITERATOR_T,
typename T>
T Sum(ITERATOR_T first, ITERATOR_T
last, T init)
{
    auto total = init;
    while (first != last) {
        total += *first;
        ++first;
    }
    return total;
}
```

- (a) `std::for_each`
  - (b) `std::merge`
  - (c) `std::accumulate`
  - (d) `std::partial_sum`
  - (e) `std::count`
2. Which of the following algorithms require that their iterator parameters point at sorted data structures?
- (a) `is_sorted`
  - (b) `numeric`
  - (c) `set_union`
  - (d) `max`
  - (e) `find`
  - (f) None of the above require sorted data structures.
3. How many parameters does the function (named or lambda) that `std::accumulate` takes as its fourth parameter have? For example, how many parameters does `func` have in the following example?

```
std::accumulate(x.begin(), x.end(),
11, func);
```

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) 4
- (f) Depends on the type of of the container (`x`).

4. What type can `x` be if `x->begin()` is not a type error?

- (a) A pointer to a `vector<long>`.
- (b) A pointer to a `set<long>`.
- (c) A `iterator` of a `vector<vector<long>>`.
- (d) A `iterator` of a `set<set<long>>`.
- (e) (a) and (b)
- (f) (a) and (c)
- (g) (c) and (d)
- (h) (b) and (d)
- (i) All of the above.
- (j) None of the above.

5. Why should you use `abs` from the `std` namespace instead of just `abs`?

- (a) Because `std::abs` will be from the `<cmath>` library instead of from `<iostream>`.
- (b) Because `std::abs` works with integers and the other works with floating point numbers.
- (c) You can use either as the standard library ensures they are the same.
- (d) Because `std::abs` is guaranteed to have all the overloaded versions.

6. What is `errno` from the `<errno>` library?

- (a) It is a type of exception raised by functions in the `<cmath>` library.
- (b) It represents the range of possible values a function can except and still not results in an overflow error.
- (c) It is the number of errors (usually caused from dividing by zero) that have occurred in the program.
- (d) It is a variable that is assigned to when a mathematical function causes some sort of error.

7. Why should you NOT just use `std::random_device` as your primary random engine?
- (a) Because it will cause the program to always generate the same random numbers.
  - (b) Because it can only be used after a pseudorandom number engine has been initialized with a seed.
  - (c) It can be used as the primary engine (and in fact should be used that way).
  - (d) Because it will cause all the other random engines to synchronize with itself.
  - (e) Because repeated use will cause it to run out of randomness.
8. `mt19937_64` is an example of what type of class?
- (a) Distribution
  - (b) Pseudorandom Number Engine
  - (c) Non-deterministic Random Number Engine
  - (d) Random number Generator
  - (e) Seed
  - (f) Random Device
9. When should you use a `std::array` instead of `std::vector`?
- (a) When you know the desired size of the container at compile time.
  - (b) When you need to store numerical values efficiently.
  - (c) When you need iterators to both the beginning and end of the container.
  - (d) When you need to store values dynamically.
  - (e) When you need to use it with an algorithm.
  - (f) When all of the values must be sorted in ascending order.
10. The value of `std::numeric_limits<double>::min()` on my computer is `2.22507e-308`. What does that mean?
- (a) That the smallest possible double is about  $2 * 10^{-308}$ .
  - (b) That the smallest difference between any two negative doubles is about  $2 * 10^{-308}$ .
  - (c) That the minimum value of any container that can hold a double is about  $2 * 10^{-308}$ .
  - (d) That the smallest positive value that a double can hold is about  $2 * 10^{-308}$ .
  - (e) All of the above are true.

