CMSC216: Bonus Review 1A

Chris Kauffman

Last Updated:
Fri Sep 26 04:49:54 PM EDT 2025

Bonus Review Rules

- 3 Questions will be shown with about 5min per question, 15min total, time limit enforced on Gradescope Quiz
- Individual student bonus dots will be calculated as

```
BonusDots = floor(log2(TotalCorrectSectionAnswers))
            - YourIncorrectAnswers
```

- Cooperation is allowed and encouraged with in your discussion section: the more correct answers in the section, the more bonus points for all
- Staff will try to facilitate discussion but will not comment on correct/incorrect answers during the quiz
- Scores will posted after all sections have taken the done the bonus review, likely the following day
- Student in the Discussion Section with the highest `TotalCorrectSectionAnswers` will get +2 BonusDots
- Bonus Review is Open Resource just like the exam: https://www.cs.umd.edu/~profk/216/exam-rules.pdf

Staging

- Open up the Gradescope Bonus Review Quiz for the day
- Once started, the quiz closes after 15min
- Get your resources set for the quiz

Okay...

Question 1

Consider the following code.

```
int a = 5;
int *ptr2 = &a;
int *ptr1 = ptr2;
```

Which best describes this code?

- (A) Dereferences `ptr2` and assigns the value `5` to `ptr1`
- (B) Assigns `ptr1` to point at the same place as `ptr2`, at variable `a`
- (C) A compile-time syntax error: a deref `*` is needed on the right hand side of the last line.
- (D) A comple-time syntax error: an address-of `&` is needed on the right hand side of the last line

Question 2

Consider this function.

```
int node_check_data(node_t *node, char *query){
  if( strcmp(node.data, query)==0 ){
    return 1;
  }
  else{
    return 0;
  }
}
```

Which best describes this code?

- (A) Incorrect use of "dot", should be `node->data` because its a pointer to a struct
- (B) Incorrect use of node, need to reference it when using it as in `*node.data`
- (C) Incorrect use of query, need to dereference it like `*query` when using it
- (D) Code is correct, no problems present

Question 3

Consider the following code.

```
{
  char str[64] = "You spoony bard!";
  ((int *) str)[2] = 1600085855;
  printf("%s\n",str);
}
```

Which best describes this code?

- (A) Will not compile due to syntax errors.
- (B) Will compile but has undefined behavior.
- (C) Will output `Yo1600085855 spoony bard!`
- (D) Will output `You spoo1600085855y bard!`
- (E) Will output `Yo____oony bard!`
- (F) Will output `You spoo____ard!`