

CSE 333 – Winter 2013 Midterm

Please do not read beyond this cover page until told to start.

A question involving what could be either C or C++ is about C, unless it explicitly states that it is about C++.

For brevity, we show only code fragments. We show only those lines of code that are pertinent to the question. If we ask “does this code compile,” you should assume that obviously required additional code (like a procedure declaration around the executable lines of code) in fact exists, even if it isn't shown (unless the questions explicitly states that it is showing all the code).

Code is considered to have compiled if no errors are raised; the code may generate warnings, but still compile.

As we all know, a pointer bug in C may or may not cause a run time error. When a question asks if there is a run time error, it means “does the code have a bug,” whether or not some execution of it produces correct output.

Name: _____

1. [3 points]

What happens when you try to compile and run this C code:

```
#include <stdlib.h>

if ( EXIT_SUCCESS ) printf("then");
else printf("else");
```

- (1) It doesn't compile
- (2) It compiles but gets a runtime error before printing anything
- (3) It prints "then"
- (4) It prints "else"

2. [4 points]

What happens when you try to compile and run this C code:

```
#include <stdio.h>
int x [] = {0, 1, 2, 3, 4, 5};
for ( int *p = &x[3]; *p; p-- ) {
    printf("%d ", *p);
}
```

- (1) It doesn't compile
- (2) It compiles but gets a runtime error before printing anything
- (3) It prints "3 2 1 "
- (4) It prints "3 2 1 ", but after that the behavior is undefined

3. [3 points]

Assume this C code compiles without warnings or errors:

```
char ** stringVec;
int y = abc(&stringVec, 0);
```

Give a plausible function prototype (declaration) for function abc.

4. [3 points]

I attempt to build an executable from a single C source file that contains just the line:

```
int x = -1;
```

If it builds, I then run it.

What happens:

- (1) It doesn't compile.
- (2) It compiles, but doesn't link.
- (3) It compiles and links, but crashes as soon as it is launched.
- (4) It compiles, links, and runs, but produces no output.
- (5) It compiles, links, runs, and prints -1.

5. [6 points]

There are two major problems with this C code. What are they? (Give a brief description of each.)

```
typedef struct string_st {
    char* data;
    unsigned int size;
} String;

// returns 1 for success, 0 for failure
int set( String s, const char* newStr) {
    if ( newStr == NULL ) return 0;
    s.size = strlen(newStr);
    s.data = (char*)malloc(s.size * sizeof(char));
    memcpy(s.data, newStr, s.size);
    return 1;
}
```

6. [3 points]

Consider this code:

```
int x = 0;
int *p = &x;
foo(p);
```

After foo returns, what are the values of x and p?

- (1) x is definitely 0; p is definitely &x
- (2) x is definitely 0; p may or may not be &x
- (3) x may or may not be 0; p is definitely &x
- (4) x may or may not be 0; p may or may not be &x

7. [2 points]

What does this C code print when compiled on a 64-bit system?

```
uint64_t a[10];
printf("%d", (int)sizeof(a));
```

- (1) 4
- (2) 8
- (3) 16
- (4) 40
- (5) 80

8. [3 points]

I try to build and run this C code. What happens?

```
typedef struct { int x; } Base;
typedef struct { int y; int z; } Sub;

Sub example = { 10, 20 };
Base* pBase = (Base*)&example;
printf("%d", pBase->x);
```

- (1) It doesn't compile
- (2) It compiles but doesn't run
- (3) It prints 0
- (4) It prints 10
- (5) It prints an unpredictable value

9. [4 points]

I try to build this C code, which occurs inside `main()`. If it builds, I then try to run it using the command line shown below the code. What happens?

```
char *strPtr;
char *outPtr = strPtr;
for (int i=1; i<argc; i++) {
    for ( char *src = argv[i]; *src != '\0'; src++ ) {
        *outPtr++ = *src;
    }
}
*outPtr = '\0';
printf("%s", strPtr);
```

```
$ ./a.out one two three
```

- (1) It doesn't compile
- (2) It compiles but has a serious run time bug
- (3) It prints "two three"
- (4) It prints "twothree"
- (5) It prints "one two three"
- (6) It prints "onetwothree"

10. [3 points]

Suppose a C implementation of a **circular, doubly linked** list includes this creation routine:

```
typedef struct llist_st {
    struct llist_st *next;
    struct llist_st *prev;
    void *data;
} llist;
llist createList() {
    llist head;
    head.next = head.prev = &head;
    return head;
}
```

A new list might be created by:

```
llist list_head = createList();
```

Briefly explain why this creation routine doesn't work.

11. [6 points]

Reminder: C++ class methods are not virtual unless explicitly declared to be so.
Here is a C++ definition of a class and a subclass.

```
class Base {
public:
    int method1();
    virtual void method2();
};
class Sub : public Base {    // Sub extends Base
public:
    int method1();
    virtual void method2();
};
```

Finally, here's some setup code:

```
Base base;
Sub sub;
Base *pBase = &sub;
```

Below is list of method invocations. For each, indicate which class fields the call, using an index from this list:

- (1) compile time error
- (2) The statement invokes the method in class **Base**
- (3) The statement invokes the method in class **Sub**

(a) base.method1()

(b) base.method2()

(c) sub.method2()

(d) pBase->method1()

(e) pbase->method2()

(f) pBase->toString()

12. [4 points]

I try to build and run the following C++ code. What happens?

```
void swap( int * &x, int * &y) {
    int *temp = x;
    x = y;
    y = temp;
}
int main(int argc, char *argv[]) {
    int x = 10;
    int y = 20;
    int *pX = &x;
    int *pY = &y;
    swap(pX, pY);
    printf("%d %d %d %d", x, y, *pX, *pY);
    return 0;
}
```

- (1) The code doesn't compile
- (2) The code compiles but gets a runtime error
- (3) It runs and prints "10 20 10 20"
- (4) It runs and prints "20 10 10 20"
- (5) It runs and prints "10 20 20 10"
- (6) It runs and prints "20 10 20 10"

13. [4 points]

This is a two part question about the following tiny bit of C++ code:

```
vector<string> vec = { "zero", "one" };
vec.push_back("two");
```

(A) When I try to compile this code (or, actually, the fully written out code that includes this code), I get these messages:

```
$ g++ -Wall -std=gnu++0x q13.cc
q13.cc: In function 'int main()':
q13.cc:4:3: error: 'vector' was not declared in this scope
q13.cc:4:10: error: 'string' was not declared in this scope
q13.cc:4:10: note: suggested alternative:
/usr/include/c++/4.6/bits/stringfwd.h:65:33: note:
'std::string'
q13.cc:4:18: error: 'vec' was not declared in this scope
```

Briefly explain how to fix the code to get rid of these errors.

(B) Suppose I've corrected all problems, so the code now builds. Write additional C++ code (that immediately follows the (possibly corrected) lines shown above) to print the contents of `vec`, one string per output line.