

CS 241

Data Organization

Quiz 2

February 1, 2018

Question 1: Automatic variable

In the C Programming Language, an *automatic variable* is:

- A A local variable in a function which comes into existence at the time the function is called, and disappears when the function is exited.
- B A variable that is automatically initialized.
- C A global variable that is automatically available to all functions within the source file.
- D A global variable that is available to all functions within any source file that declare the variable as extern.
- E A variable that is automatically defined by the compiler such as PI, E, and HBAR.

Question 1: Automatic variable

In the C Programming Language, an *automatic variable* is:

- A A local variable in a function which comes into existence at the time the function is called, and disappears when the function is exited.
- B A variable that is automatically initialized.
- C A global variable that is automatically available to all functions within the source file.
- D A global variable that is available to all functions within any source file that declare the variable as extern.
- E A variable that is automatically defined by the compiler such as PI, E, and HBAR.

Question 2: if, else if, else

```
int main(void)
{
    int x = 4;

    if (x == 1)
    {
        printf("x is 1\n");
    }
    else if (x == 2)
    {
        printf("x is 2\n");
    }
    else x = 3;
    {
        printf("x is %d\n", x);
    }
}
```

What is the output of this code?

A x is 1

B x is 2

C x is 3

D x is 4

E Nothing is printed.

Question 2: if, else if, else

```
int main(void)
{
    int x = 4;

    if (x == 1)
    {
        printf("x is 1\n");
    }
    else if (x == 2)
    {
        printf("x is 2\n");
    }
    else x = 3;
    {
        printf("x is %d\n", x);
    }
}
```

What is the output of this code?

A x is 1

B x is 2

C x is 3

D x is 4

E Nothing is printed.

Question 3: Functions

This code will not compile because:

```
1  int foo(float x);  
2  
3  void main(void)  
4  {  
5      int n=5;  
6      printf("%d\n", foo(n));  
7  }  
8  
9  int foo(int n)  
10 {  
11     return 2*n;  
12 }
```

- A The version of foo in line 1 accepts a float, but returns an int.
- B The function foo in line 1 has no body.
- C The version of foo in line 1 should not end with a semicolon.
- D The variable n is declared in two different places.
- E The prototype of foo does not agree with the definition.

Question 3: Functions

This code will not compile because:

```
1  int foo(float x);
2
3  void main(void)
4  {
5      int n=5;
6      printf("%d\n", foo(n));
7  }
8
9  int foo(int n)
10 {
11     return 2*n;
12 }
```

- A The version of foo in line 1 accepts a float, but returns an int.
- B The function foo in line 1 has no body.
- C The version of foo in line 1 should not end with a semicolon.
- D The variable n is declared in two different places.
- E The prototype of foo does not agree with the definition.