CS 241
Data Organization
Quiz 2

January 30, 2018

Question 1: Syntax

Which of these statements would result in an error?

A
```
int a = (float) 3 / (float) 4;
```

B
```
int b = (int) 'c';
```

C
```
int c = 1/2 + 1/3 + 1/4 + 1/5
```

D
```
int d = (5 - 2) > 3;
```

E
```
int e = 5%17;
```

Question 1: Syntax

Which of these statements would result in an error?

A
```
int a = (float) 3 / (float) 4;
```

B
```
int b = (int) 'c';
```

C
```
int c = 1/2 + 1/3 + 1/4 + 1/5
```

D
```
int d = (5 - 2) > 3;
```

E
```
int e = 5%17;
```

Question 2: '=' symbol

In the C programming language, the '=' symbol is most accurately read:

A "Equals"

B "Variable Assignment"

C "Is equivalent to"

D "A mathematical symbol used to indicate equality"

E "A conditional symbol used to indicate equality"

Question 2: '=' symbol

In the C programming language, the '=' symbol is most accurately read:

A "Equals"

B "Variable Assignment"

C "Is equivalent to"

D "A mathematical symbol used to indicate equality"

E "A conditional symbol used to indicate equality"

Question 3: Call by value

In the C Programming Language, *call by value* means:

A When two functions have the same name, the compiler determines which to call by the value of the arguments.

B When calling a function, the called function is given the address of its arguments so that the function can both read and set the arguments values.

C Each called function is assigned a value that is used by the operating system to determine the functions priority. This is most useful on multi-core systems.

D When calling a function, the called function is given the values of its arguments, copied into temporary variables.

E Before a function is evaluated, the arguments are substituted directly into the function. Only when the function is evaluated, then the function evaluates the values.

Question 3: Call by value

In the C Programming Language, *call by value* means:

A. When two functions have the same name, the compiler determines which to call by the value of the arguments.

B. When calling a function, the called function is given the address of its arguments so that the function can both read and set the arguments values.

C. Each called function is assigned a value that is used by the operating system to determine the functions priority. This is most useful on multi-core systems.

D. When calling a function, the called function is given the values of its arguments, copied into temporary variables.

E. Before a function is evaluated, the arguments are substituted directly into the function. Only when the function is evaluated, then the function evaluates the values.

Question 4: if, else if, else

```
int main(void){
  int x = 10;
  if (x == 10)
  {
     printf("x is 10\n");
   }
   else if (x = 2);
    {
      printf("x is 2\n");
    }
   printf("x is %d\n", x);
   return 0;
}
```

What is the output of this code?

A x is 10

B x is 10 x is 2 x is 2

C x is 10 x is 10

D x is 10 x is 2 x is 10

Question 4: if, else if, else

```
int main(void){
   int x = 10;
   if (x == 10)
   {
      printf("x is 10\n");
    }
    else if (x = 2);
     {
       printf("x is 2\n");
     }
    printf("x is %d\n", x);
    return 0;
}
```

What is the output of this code?

A x is 10

B x is 10 x is 2 x is 2

C x is 10 x is 10

D x is 10 x is 2 x is 10