Reading Code

TOTAL POINTS 10

1. What is an *lvalue*? 1 point

- A value that can be placed into a box.
- A value that can be the left operand of an operator.
- Something that names a box.
- A value that is elevated through the evaluation of an expression.

2. What does *scope* mean? 1 point

- The kinds of problem that a function can solve.
- The amount of domain knowledge required to write a particular program.
- The types of values that a variable can hold.
- The region of code in which a variable is visible.

3. Suppose you have the line of code: 1 point

```
int a = f(x,y);
```

Which of the following best describes how you determine what value to put in the box for **a**?

- 1. Look at **f** to see what mathematical function it is.
  2. Work out the math for **x** and **y**.
  3. Your answer goes in the box for **a**.
- 1. Create a frame for **f**, copying the values of **x** and **y** into the boxes named for its parameters.
  2. Move the execution arrow into **f**, and execute code line by line.
  3. When your execution arrow reaches a statement of the form **return expression;** the value of that expression is what you will end up putting in the box for **a** (after you destroy the frame, and return the execution arrow to the call site).
- 1. Create a frame for **f**, copying the values of **x** and **y** into the boxes named for its parameters.
  2. Move the execution arrow into **f**, and execute code line by line.
  3. When your execution arrow reaches a statement of the form **printf("%d", num)** the value that it prints is what goes into the box for **a**.
- 1. Look at **f** to see where it has a statement of the form **a = expression**
  2. Figure out what value that expression has
  3. The value you came up with is what goes in the box for **a**.

4. If C did not have the keyword "for" but you wanted to write something where a for-loop were the natural choice, what could you use instead? 1 point

- **return**
- **break**
- **while**
- **if**

5. What is the difference between printing a value and returning a value? 1 point

- Printing a value leaves the current function, while returning a value does not.
- Printing a value only works on strings, while returning a value only works on integers.
- Printing a value gives it to the user, while returning a value gives it to other code for further computation.

6. For the following erroneous code: 1 point

```
int f (int x) {
  int answer = 0
  for (int i = 0; i < X; i++) {
    answer += i * i;
  }
  return answer;

```

What is the error on line 2?

- type name missing from variable declaration
- missing {
- missing ;
- undeclared variable

7. Execute the following code by hand: 2 points

```
1  int main (void) {
2    int a = 3;
3    int b = 6;
4    while (a <= b) {
5      if (a % 2 == 1) {
6        printf("a is %d\n", a);
7      }
8      else {
9        printf("b is %d\n", b);
10       for (int i = 0; i < b - a ; i++) {
11         printf("a * i + b = %d\n", a * i + b);
12       }
13     }
14     a++;
15     b--;
16   }
17   return 0;
18 }
```

Which one of the following gives the correct output?

- a is 3

  b is 5

- b is 6

  a * i + b = 6

  a * i + b = 9

  a * i + b = 12

  a is 4

- a is 3

  b is 5

  a * i + b = 5

  a * i + b = 9

- b is 6

  a * i + b = 6

  a is 4

- a is 3

  b is 5

  a * i + b = 5

8. Execute the following code by hand: 2 points

```
1  int anotherFunction(int a, int b) {
2    int answer = 2;
3    int x = 0;
4    printf("In anotherFunction(%d,%d)\n",a,b);
5    while (b > a) {
6      printf("a is %d, b is %d\n", a, b);
7      answer = answer + (b - a);
8      b -= x;
9      a += x / 2;
10     x++;
11   }
12   return answer;
13 }
14
15 int someFunction(int x, int y) {
16   int a = x + y;
17   if (x < y) {
18     for (int i = 0; i < x; i++) {
19       printf("In the loop with i = %d, a = %d\n", i, a);
20       a = a + x;
21     }
22   }
23   else {
24     y = anotherFunction(y,a+1);
25   }
26   return a * (y-10);
```

```
}

int main(void) {
  int x = 2;
  int b = someFunction(3,x);
  printf("b = %d\n", b);
  printf("x = %d\n", x);
  return 0;
}
```

Which one of the following gives the correct output?

- In anotherFunction(3,2)

  b = 2

  x = 2

- In the loop with i = 0, a = 5

  In the loop with i = 1, a = 7

  b = -63

  x = 2

- In the loop with i = 0, a = 5

  In the loop with i = 1, a = 8

  In the loop with i = 2, a = 11

  b = -112

  x = 2

- In anotherFunction(2,6)

  a is 2, b is 6

  a is 2, b is 6

  a is 2, b is 5

  b = 15

  x = 2

I, **Sachin Kumar**, understand that submitting another's work as my own can result in zero credit for this assignment. Repeated violations of the Coursera Honor Code may result in removal from this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Save Submit