Yo, is this going to be on the test?
- Student

Dear Student:
It may or may not be on the test, but it will be applicable to your future career, for which there is no test.
When you get paid to do a job, they expect you to know this stuff!
- Your Instructor

You never know when it’s going to come in handy.
Study for life, not just for this test.
- Your Instructor

Uh, is that a “yes” or a “no”?

It’s a “yo.”
Study Sessions

• Monday, 6-7:30 pm, WNGR 116 (70 people out of 100 possible)
• Tuesday, 6-7:30 pm, WNGR 116 (80 people out of 100 possible)
Week 1: Variables and Basics

• A variable that can hold a whole number is called a(n) **integer**.
• A digit that can hold a zero or a one is known as a **bit**.
• Errors in a program can be classified into three types, list them **Syntax, Logic, Runtime**
• A mistake that is a direct violation of the syntax rules will generate a compiler **error**.
• `int myValue;` is called a **variable declaration**.
Week 1 Continued

• A memory address is
  a) Where a variable is stored
  b) Where the computer is located
  c) A step in the program.
  d) Where the CPU is stored.

• What does the following line of code display to the screen?
  a) cout << “This is the computer
programming book”;
  b) This is the computern programming book
  c) This is the computer
  d) Nothing
  e) This is the computer
     programming book
Week 1 Continued

• `cout << “How many items would you want?\n”;
  a) is an output statement
  b) is an input statement
  c) is a variable declaration
  d) is a program

• `#include <iostream>`
  a) is a variable declaration
  b) an executable statement
  c) an include directive
  d) illegal code
Week 1 Continued

• What is wrong with the following statement?
  cout << “Hello to everyone\n”
  a) cout should be count
  b) missing a semicolon
  c) missing a “
  d) missing a ( 

• True or False: The compiler will catch all your programming mistakes.
Week 1 Continued

• What is the output of the following code?

```c
float value;
value = 33.5;
cout << "value" << endl;
```

a) 33.5  
b) 33  
c) value  
d) garbage
Week 1 Continued

• What is the value of \( x \) after the following statements?

```c
float x;
x = 15/4;
```

a) 3.75  
b) 4.0  
c) 3.0  
d) 60
Week 2: Conditionals

• if-else statements that are inside other if-else statements are said to be **nested**.

• When must we use braces to define the body of a conditional expression? **more than one line**

• In a compound logical and (&&) expression, the evaluation of the expression stops once one of the terms of the expression is false. This is known as **short-circuiting** evaluation.

• The code following the **default** case is executed if none of the other cases are matched in a switch statement.
Week 2 Continued

• Given the following code fragment and the input value of 4.0, what output is generated?

    float tax;
    float total;
    cout << "enter the cost of the item\n";
    cin >> total;
    if ( total >= 3.0) {
        tax = 0.10;
        cout << total + (total * tax) << endl;
    } else {
        cout << total << endl;
    }

    a) 3
    b) 3.3
    c) 4.0
    d) 4.4
Week 2 Continued

• If x has the value of 3, y has the value of -2, and w is 10, is the following condition true or false?
  
  \[
  \text{if( } x < 2 \text{ && } w < y \text{)}
  \]

  a) true
  
  b) false

  \[
  3 \ F \quad 10 \ < \ -2
  \]

  \[
  \quad \ F \ \ A \quad T
  \]

• What is the correct way to write the condition \( y < x < z \)?

  a) \( y < x < z \)

  b) \( (y < x) \text{ && } z \)

  c) \( ((y > x) \text{ || } (y < z)) \)

  d) \( ((y < x) \text{ && } (x < z)) \)
Week 2 Continued

- Given the following code fragment, and an input value of 3, what is the output that is generated?

```cpp
int x;
cout << "Enter a value\n"; // evaluates the x = 0
cin >> x;
if(x==0) {
    cout << "x is zero\n"; // first, then examines the value of x
} else {
    cout << "x is not zero\n"; // since x is zero, it will return false
}
```

- a) x is zero
- b) x is not zero
- c) unable to determine
- d) x is 3

If it was a non-zero number, it would be true.
Week 2 Continued

• Given the following code fragment, what is the output?

```cpp
int x=5;
if( x > 5)
    cout << "x is bigger than 5. ";
    cout <<"That is all. ";
    cout << "Goodbye\n";
```

a) x is bigger than 5. That is all
b) x is bigger than 5
c) That is all. Goodbye
d) Goodbye
Week 2 Continued

• Which of the following are valid case statements in a switch?
  a) case 1:
  b) case x<4:
  c) case 'ab':
  d) case 1.5:

  Bad question
Week 3: Loops

• True or False: The body of a do-while loop always executes at least once.  **True**

• True or False: Loops are used when we need our program to make a choice between two or more things.  **False**

• Each repetition of a loop body is called **Iteration**.

• A loop that iterates one too many or one too few times is said to be **off by one**.
Week 3 Continued

• Given the following code fragment, what is the final value of y?

```cpp
int x, y;
x = -1;
y = 0;
while(x <= 3) {
    y += 2;
    x += 1;
}
```

- a) 2
- b) 10
- c) 6
- d) 8

\[ y = -1 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \]
\[ y = 0 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 8 \rightarrow 10 \]
Week 3 Continued

• What is the final value of x after the following fragment of code executes?
  ```java
  int x=0;
  do {
    x++;
  } while(x > 0);
  ```
  a) 8
  b) 9
  c) 10
  d) 11
  e) infinite loop.
Week 3 Continued

• Given the following code, what is the final value of i?

```cpp
int i;
for(i=0; i<=4;i++) {
    cout << i << endl;
}
```

a) 3  
b) 4  
c) 5  
d) 0
Week 3 Continued

• Given the following code, what is the final value of i?

```c
int i,j;
for(i=0;i<4;i++) {
    for(j=0;j<3;j++) {
        if(i==2)
            break;
    }
}
```

a) 3  
b) 4  
c) 5  
d) 0
Week 3 Continued

• Which of the following is not a good reason for choosing a certain loop control?
  a) What the loop does
  b) The minimum number of iterations of the loop
  c) The condition for ending the loop
  d) If the loop is in a function
Week 3 Continued

• What is wrong with the following for loop?

```c++
for(int i=0;i<10;i--) {
    cout << "Hello\n";
}
```

a) can not use a for-loop for this
b) i is not initialized

\[\text{C} \]

\[\text{c} \]

\[\text{infinite loop} \]

d) off-by-one error
Week 4: Functions

• Variables defined inside a set of braces are said to be **local** to that block of code.
• True or False: A function may return more than one item.
• True or False: Function naming rules follow variable naming rules.
• True or False: The types of parameters are optional in the function declaration.
• True or False: It is possible to have a function that has no parameters.
• True or False: The parameters listed in the function declaration are considered global variables.
• True or False: $\text{pow}(2,3)$ is the same as $\text{pow}(3,2)$. 
Week 4 Continued

• In the following function declaration, the variable size is known as a **parameter**.
  
  int myFunction ( int size);

• The **function body** describes how the function will work.

• The **scope** of a variable is where that variable can be used.
Week 4 Continued

• What is the value returned by the following function?

```c
int function() {
    int value = 35;
    return value + 5;
    value += 10;
}
```

a) 35  b) 40  c) 50  d) 10
• When overloading a function, what must be true?
  a) The names should be different with the same number and/or types of parameters.
  b) The names should be the same with different number and/or types of parameters.
  c) The names should be different with different number and/or types of parameters.
  d) The names should be the same with the same number and/or types of parameters.
• Which of the following are valid function calls to the fabs function?
  a)  \texttt{fabs(3.5)};
  b)  \texttt{cout \ll \text{fabs(3.5)}};
  c)  \texttt{cin \gg \text{fabs(3.5)}};
  d)  \texttt{fabs(cin \gg x)};
  e)  a,b and c
  f)  a and b
Week 4 Continued

• Multiple arguments to a function are separated by
  a) comments
  b) semicolons
  c) colons
  d) commas
  e) periods
Week 4 Continued

• What is the value of i after the following function call?

```c
int doSomething(int value) {
    value = 35;
    return value;
    value = 13
}
```

//fragment of main program
int i=0;
cout << doSomething(i);

a) 13
b) 35
c) 48
d) 0