CSCI 200: Foundational Programming Concepts & Design

Exam I Review

Procedural Programming Quiz

- Make Canvas Full Screen
- Access Code:
- 12 Minutes

1. What is the result?

a) 2 + 3 * 4 – 6

b) 5 + 11 / 3

c) 11 % 3 * 4

d) (2 + 1) * 3 - 1

1. What is the result?

a) 2 + 3 * 4 – 6 8

b) 5 + 11 / 3 8

c) 11 % 3 * 4 8

d) (2 + 1) * 3 - 1 8

2. Create boolean test conditions

a) myHeight is greater than 2

b) y is odd and less than 10

c) At least one of x or y is 3

d) t is between 2.1 and 2.3 inclusive

2. Create boolean test conditions

a) myHeight is greater than 2

```
myHeight > 2
```

b) y is odd and less than 10

```
y % 2 == 1 && y < 10
```

c) At least one of x or y is 3

```
x == 3 || y == 3
```

d) t is between 2.1 and 2.3 inclusive

```
t >= 2.1 && t <= 2.3
```

3. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 12;

    if( (x >= 2) || (x != 17) )
        cout << x << endl;
    else
        cout << "Have a good day" << endl;

    return 0;
}
```

3. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 12;

    if( (x >= 2) || (x != 17) )
        cout << x << endl;
    else
        cout << "Have a good day" << endl;

    return 0;
}
```

12

4. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 1;

    if( (x >= 2) || (x != 17) )
        cout << x << endl;
    else
        cout << "Have a good day" << endl;

    return 0;
}
```

4. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 1;

    if( (x >= 2) || (x != 17) )
        cout << x << endl;
    else
        cout << "Have a good day" << endl;

    return 0;
}
```

1

5. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 17;

    if( (x >= 2) && (x != 17) )
        cout << x << endl;
    else
        cout << "Have a good day" << endl;

    return 0;
}
```

5. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 17;

    if( (x >= 2) && (x != 17) )
        cout << x << endl;
    else
        cout << "Have a good day" << endl;

    return 0;
}
```

Have a good day

6. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 11, y = 5;

    int answer;
    answer = x / y;
    cout << answer << endl;

    return 0;
}
```

6. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 11, y = 5;

    int answer;
    answer = x / y;
    cout << answer << endl;

    return 0;
}
```

2

7. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 9, y = 2;

    cout << x / y << endl;
    cout << (double)x / (double)y << endl;
    cout << (double)x / y << endl;
    cout << x / (double)y << endl;

    return 0;
}
```

7. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 9, y = 2;

    cout << x / y << endl;
    cout << (double)x / (double)y << endl;
    cout << (double)x / y << endl;
    cout << x / (double)y << endl;

    return 0;
}
```

4

4.5

4.5

4.5

8. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 5, y = 10;

    y = x++;
    cout << x << " " << y << endl;

    y = ++x;
    cout << x << " " << y << endl;

    return 0;
}
```

8. What is the output?

```
#include <iostream>
using namespace std;

int main() {
    int x = 5, y = 10;

    y = x++;
    cout << x << " " << y << endl;

    y = ++x;
    cout << x << " " << y << endl;

    return 0;
}
```

6 5

7 7

9. Find the Errors

```
#include <iostream>
using namespace std

int main() {
    int x = 6;
    double y = 2.5;
    z = 1;
    cin << z;
    if( x = y )
        cout "x and y match";
    else
        cout "x and y do not match";

    return 0;
}
```

9. Find the Errors

```
#include <iostream>
using namespace std;

int main() {
    int x = 6;
    double y = 2.5;
    int z = 1;
    cin >> z;
    if( x == y )
        cout << "x and y match";
    else
        cout << "x and y do not match";

    return 0;
}
```

10. Write if/else code

a) Write a series of if statements (use only if) that will output a student's letter grade based on the input. Assume the input (already received) is called examScore and that the value of examScore is greater than 70 and less than 100.

b) Write an if block (if and else if) that will output a student's letter grade based on the input. Assume the input (already received) is called examScore and that the value of examScore is greater than 70 and less than 100.

10a

```
if( examScore >= 90 )
  cout << "A" << endl;
if( examScore >= 80 && examScore < 90 )
  cout << "B" << endl;
if( examScore < 80 )
  cout << "C" << endl;
```

10b

```
if( examScore >= 90 )
  cout << "A" << endl;
else if( examScore >= 80 )
  cout << "B" << endl;
else
  cout << "C" << endl;
```

11. Write Loop code

a) Write a snippet of code that prints all odd numbers between 0 and X (inclusive), where X is given by the user. Use a while loop.

b) Write a snippet of code that prints all odd numbers between 0 and X (inclusive), where X is given by the user. Use a for loop.

11a

```
int x;
cin >> x;
int i = 0;
while( i <= x ) {
  if( i % 2 )
    cout << i << endl;
  i++;
}
```

11b

```
int x;
cin >> x;
for( int i = 0; i <= x; i++ ) {
  if( i % 2 )
    cout << i << endl;
}
```

12. Rewrite as a switch

```
if( (rank == 1) || (rank == 2) )
    cout << "Lower division" << endl;
else {
    if( (rank == 3) || (rank == 4) )
        cout << "Upper division" << endl;
    else {
        if( rank == 5 )
            cout << "Graduate student" << endl;
        else
            cout << "Invalid rank" << endl;
    }
}
```

12. Rewrite as a switch

```
switch( rank ) {
  case 1:
  case 2:
    cout << "Lower division" << endl;
    break;
  case 3:
  case 4:
    cout << "Upper division" << endl;
    break;
  case 5:
    cout << "Graduate student" << endl;
    break;
  default:
    cout << "Invalid rank" << endl;
}
```

13. True or False

a) The statement “x++” adds one to x.

b) A semi-colon is needed at the end of a while code block.

c) Once a constant variable has been created, it cannot be changed.

d) Boolean variables store the values always true, always false, or sometimes true.

13. True or False

a) TRUE

b) FALSE

c) TRUE

d) FALSE

14. Rewrite as a for loop

a)

```
int i = 2;
while( i <= 18 ) {
    cout << "*";
    i += 3;
}
```

b) What is the output?

14. Rewrite as a for loop

a)

```
for( int i = 2; i <= 18; i += 3 ) {
    cout << "*";
}
```

b) What is the output?

14. Rewrite as a for loop

a)

```
for( int i = 2; i <= 18; i += 3 ) {
    cout << "*";
}
```

b) ******

15. What is the output?

```
int number = 0;
int sum = 0;
int limit = 20;

while( number > limit ) {
    sum += number;
    number += 2;
}

cout << sum << endl;
```

15. What is the output?

```
int number = 0;
int sum = 0;
int limit = 20;

while( number > limit ) {
    sum += number;
    number += 2;
}

cout << sum << endl;
```

0

16. What is the output?

```
int number = 100;
int sum = 0;
int limit = 20;

while( number > limit ) {
    sum += number;
    number += 2;
}

cout << sum << endl;
```

16. What is the output?

```
int number = 100;
int sum = 0;
int limit = 20;

while( number > limit ) {
    sum += number;
    number += 2;
}

cout << sum << endl;
```

Inf. Loop

17. What is the output?

```
int number = 0;
int sum = 0;
int limit = 20;

while( number < limit ) {
    sum += number;
    number += 2;
}

cout << sum << endl;
```

17. What is the output?

```
int number = 0;
int sum = 0;
int limit = 20;

while( number < limit ) {
    sum += number;
    number += 2;
}

cout << sum << endl;
```

90

18. What is the output?

```
for( int i = 0; i < 4; i++ ) {
    for( int j = i; j < 6; j++ )
        cout << "*";
    cout << endl;
}
```

18. What is the output?

```
for( int i = 0; i < 4; i++ ) {
    for( int j = i; j < 6; j++ )
        cout << "*";
    cout << endl;
}
```

```
******
*****
****
***
```

19. Random Numbers

- Write a program that prints 10 random numbers between 1 and 100. Use an appropriate seed.

19. Random Numbers

```
srand( time(0) );
rand();
for( int i = 0; i < 10; i++ )
  cout << (rand() % 100) + 1 << endl;
```

20. True or False

a) void functions return a value.

b) Function prototypes do not require parameter names.

20. True or False

a) void functions return a value. FALSE

b) Function prototypes do not require parameter names.

20. True or False

a) void functions return a value. FALSE

b) Function prototypes do not require parameter names. TRUE

21. What is printed?

```
void my_func( int x, int y ) {
    x = 52;
    y = 7;
}

int main() {
    int x = 0;
    int y = 0;
    my_func( x, y );
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    return 0;
}
```

21. What is printed?

```
void my_func( int x, int y ) {
    x = 52;
    y = 7;
}

int main() {
    int x = 0;
    int y = 0;
    my_func( x, y );
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    return 0;
}
```

x = 0

y = 0

22. What is printed?

```
int my_function( double a, double b, double c ) {
    a = 2 * b;
    b = 15 + c;
    c = 3 * a;
    return (a + b + c);
}

int main() {
    double a = 1;
    double b = 2;
    double c = 3;
    double d = my_function( a, b, c );
    cout << "d = " << d << endl;
    return 0;
}
```

22. What is printed?

```
int my_function( double a, double b, double c ) {
    a = 2 * b;
    b = 15 + c;
    c = 3 * a;
    return (a + b + c);
}

int main() {
    double a = 1;
    double b = 2;
    double c = 3;
    double d = my_function( a, b, c );
    cout << "d = " << d << endl;
    return 0;
}
```

d = 34

23. Which are legal statements?

```
void func_A( int x, int y, int z );
int func_B( int x, double y );
```

a) `cout << func_A( 5, 4, 3 ) << endl;`

b) `cout << func_B( 5, 4.0 ) << endl;`

c) `func_A( 5, 4 );`

d) `func_A( 5, 4.7, 3 );`

e) `int x = func_B( 5, 6 );`

f) `int y = func_A( 5, 4, 3 );`

23. Which are legal statements?

```
void func_A( int x, int y, int z );
int func_B( int x, double y );
```

a) `cout << func_A( 5, 4, 3 ) << endl;` NO

b) `cout << func_B( 5, 4.0 ) << endl;` YES

c) `func_A( 5, 4 );` NO

d) `func_A( 5, 4.7, 3 );` YES

e) `int x = func_B( 5, 6 );` YES

f) `int y = func_A( 5, 4, 3 );` NO

24. Write Function Prototypes

a) Write a function prototype with the name "cool_func" that has no parameters and does not return a value.

b) Write a function prototype with the name "hot_func" that has no parameters and returns a double.

c) Write a function prototype with the name "neutral_func" that returns an integer and whose parameters in order are an integer named foo, a double named bar, and a character named baz.

24. Write Function Prototypes

a) `void cool_func();`

b) `double hot_func();`

c) `int neutral_func( int foo, double bar, char baz );`

25. Write a Function

- Write a function that calculates and returns the area of a square for whole numbers.

25. Write a Function

- Write a function that calculates and returns the area of a square for whole numbers.

```
int square_area( const int SIDE_LENGTH ) {
  return SIDE_LENGTH * SIDE_LENGTH ;
}
```

26. Find the errors

```
#include <iostream>
using namespace std;

int main() {
    int 7;
    add_one( x );
    cout >> "7 plus one is " << x << endl;
    return 0;
}

void add_one( int x ) {
  ++x;
}
```

26. Find the errors

```
#include <iostream>
using namespace std;

int add_one( int x );

int main() {
    int x = 7;
    x = add_one( x );
    cout << "7 plus one is " << x << endl;
    return 0;
}

int add_one( int x ) {
  return ++x;
}
```

27. Memory Time

- With Two's Complement and Floating Point Binary Representation, what does the leading bit correspond to?

- What does this do to the number of integers that can be stored?

27. Memory Time

- The sign (positive/negative) of the value

- What does this do to the number of integers that can be stored?

27. Memory Time

- The sign (positive/negative) of the value

- Halves the max allowable number
  - e.g. 2^{31} instead of 2^{32}

28. Data Type Modifiers

- What affect to the allowable values of a standard `int` data type do the following modifiers have?
  - `unsigned long long int`
  - `long long int`

- What's the difference between the two?

28. Data Type Modifiers

- `int` (32 bits):
  - -2^{31} to $+2^{31}$
- `long long int` (64 bits):
  - -2^{63} to $+2^{63}$
- `unsigned long long int` (64 bits):
  - 0 to $+2^{64}$

29. Data Type Modifiers

- What is the output of the following code?

```
unsigned short int q = -1;
cout << "q is: " << q << endl;
```

a) -1
b) 65535
c) Compiler Error
d) Runtime Error

- Why?

29. Data Type Modifiers

- What is the output of the following code?

```
unsigned short int q = -1;
cout << "q is: " << q << endl;
```

a)

b) 65535

c)

d)

- Why?

29. Data Type Modifiers

- What is the output of the following code?

```
unsigned short int q = -1;
cout << "q is: " << q << endl;
```

a)

b) 65535

c)

d)

- Negative value assigned with leading bit, but interpreted as unsigned value

30. Building

- When compiling our C++ code into a binary object file, the object file has a larger file size than the C++ file. Why?

30. Building

- When compiling our C++ code into a binary object file, the object file has a larger file size than the C++ file. Why?

1. The binary representation is longer
2. The included library code gets inserted into our file by the compiler

31. Makefiles

- What advantages to the build process does a Makefile provide?

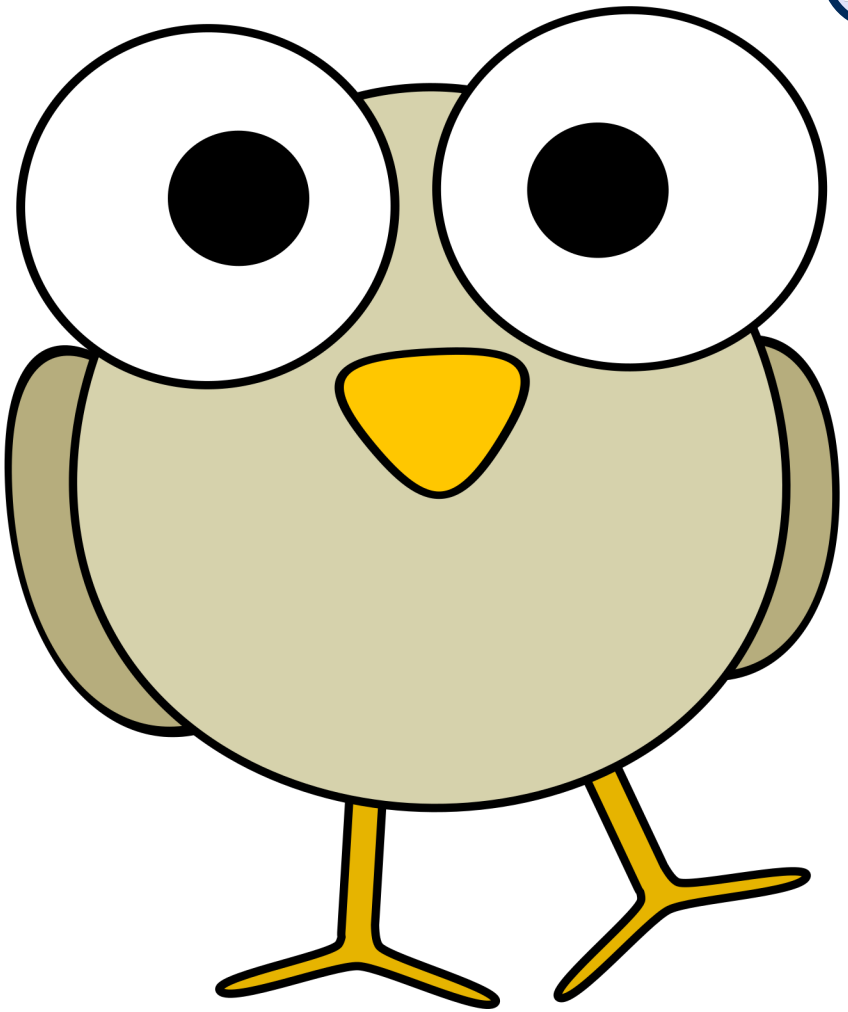
- Why do we separate out the compile and link steps of our build process?

31. Makefiles

- Faster / smarter / simpler builds for the developer

- Only rebuild the components that have changed since last build

Questions?



For Next Time

- Exam I Wednesday in Class