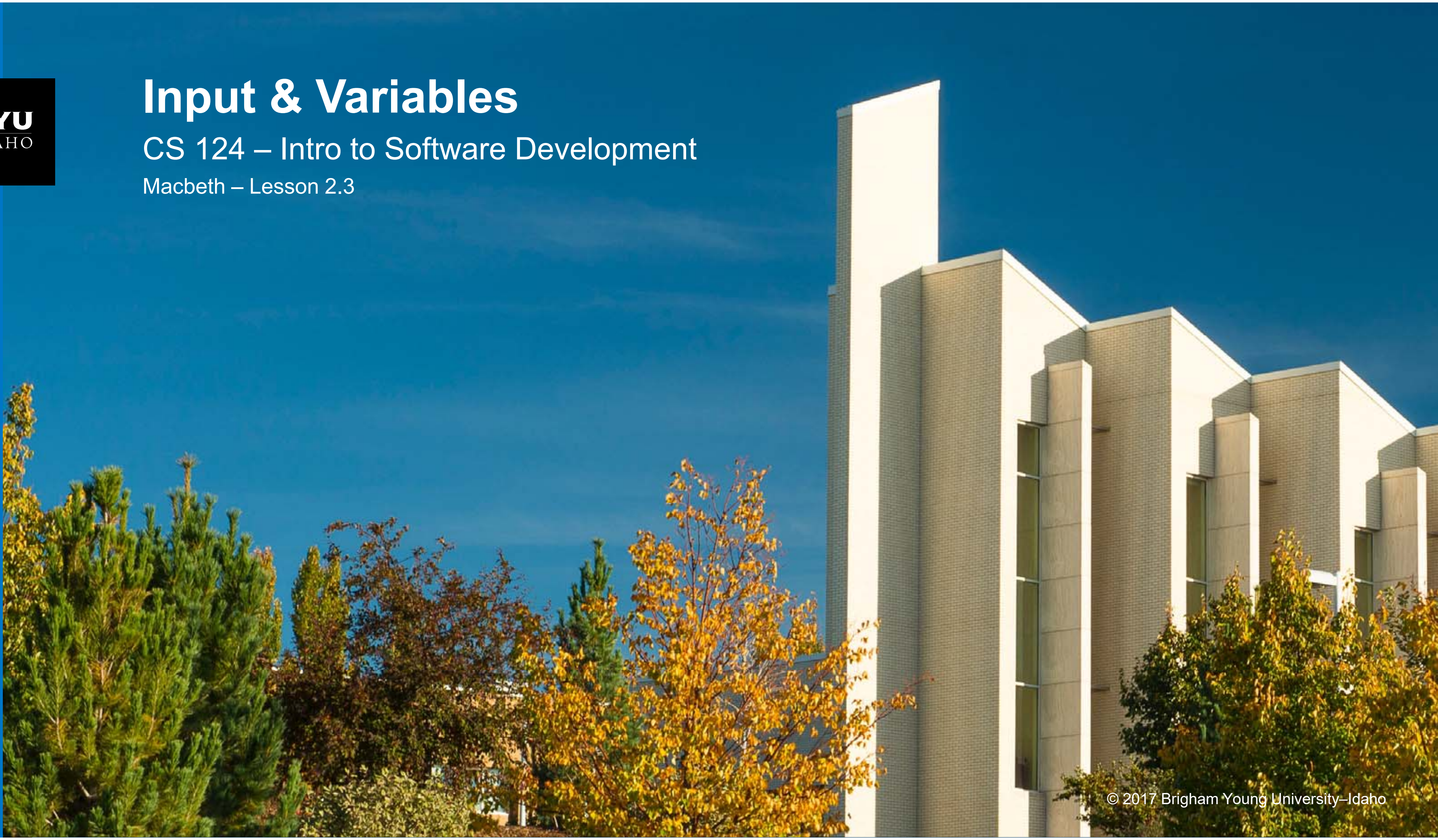
Input & Variables

CS 124 – Intro to Software Development

Macbeth – Lesson 2.3



Agenda

- Opening Prayer
- Music Friday
- Q&A
  - Review Assignment
  - Review Next Project
- Quiz Review
- Variables
- Input
- Looking Ahead

Music Friday

Nearer, My God, to Thee (Hymn 100)

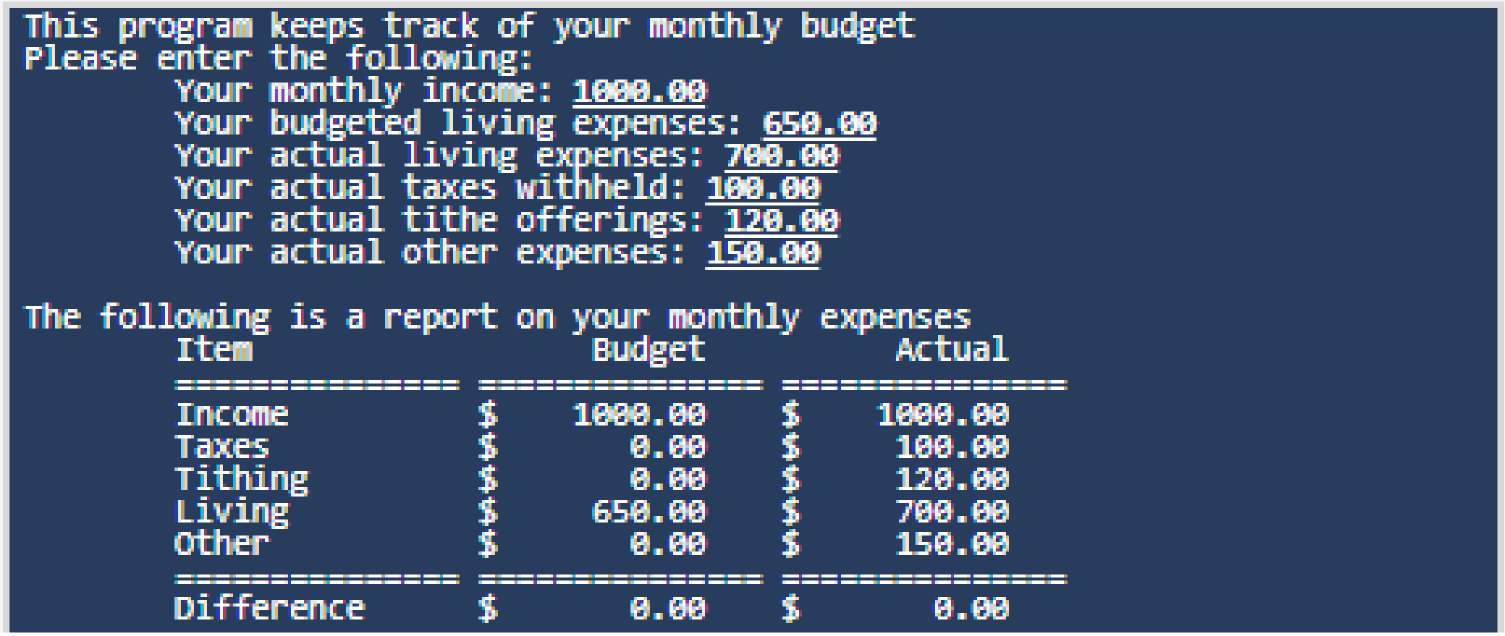
Nearer, my God, to thee,
Nearer to thee!
E'en though it be a cross
That raiseth me.
Still all my song shall be
Nearer, my God, to thee,
Nearer, my God, to thee,
Nearer to thee!

Review Last Assignment

- What was the hardest part?
- How did you solve it?

- When I share my code on the screen, please do not take pictures ☺

02 Ponder – Monthly Budget Project



```
This program keeps track of your monthly budget
Please enter the following:
        Your monthly income: 1000.00
        Your budgeted living expenses: 650.00
        Your actual living expenses: 700.00
        Your actual taxes withheld: 100.00
        Your actual tithe offerings: 120.00
        Your actual other expenses: 150.00

The following is a report on your monthly expenses
        Item                  Budget          Actual
        =============== =============== ===============
        Income          $    1000.00    $    1000.00
        Taxes           $       0.00    $     100.00
        Tithing         $       0.00    $     120.00
        Living          $     650.00    $     700.00
        Other           $       0.00    $     150.00
        =============== =============== ===============
        Difference      $       0.00    $       0.00
```

Variables

Variables are a named location where we store data. Notice all variables are written in "camelBack" notation (each word has a capital letter except the first letter).

```
int main()
{
    float circleRadius;
    float circleArea;

    cout << "What is the radius of the circle? ";
    cin >> circleRadius;

    circleArea = 3.14 * circleRadius * circleRadius;

    cout << "The area is: " << circleArea << endl;
    return 0;
}
```

Variable Types

Type	Use	Size	Range
bool	Logic	1 (8 bits)	true, false
char	Letters and Small Numbers	1 (8 bits)	-128 to 127 (2^{7} – 1)
unsigned char	Small Positive Numbers	1 (8 bits)	0 to 255 (2^{8} – 1)
short (short int)	Medium Numbers	2 (16 bits)	-32,768 to 32,767 (2^{15} – 1)
unsigned short	Medium Positive Numbers	2 (16 bits)	0 to 65,535 (2^{16} – 1)
int	Large Numbers	4 (32 bits)	-2,147,483,648 to 2,147,483,647 (2^{31} – 1)
unsigned int	Large Positive Numbers	4 (32 bits)	0 to 4,294,967,295 (2^{32} – 1)
long (long int)	Very Large Numbers	8 (64 bits)	-9,223,372,036,854,775,808 to 9,223,372036,854,775,807 (2^{63} – 1)
unsigned long	Very Large Positive Numbers	8 (64 bits)	0 to 18,446,744,073,709,551,615 (2^{64} – 1)
float	Real Numbers (decimals)	4 (32 bits)	10^{-38} to 10^{38} (accurate to 7 digits precision)
double	Larger Real Numbers	8 (64 bits)	10^{-308} to 10^{308} (accurate to 15 digits precision)
long double	Very Large Real Numbers	16 (128 bits)	10^{-4932} to 10^{4932} (accurate to 19 digits precision)

Integers

- The most common data type for storing positive and negative numbers.
- Useful for counting.
- Be careful using integers for real world data like temperatures, distances, and weights. These are usually represented as real numbers (with a decimal).

```
int numCars;
int studentID;
int ageYears;
```

```
numCars = 72110000; // Number cars built worldwide in 2016
```

Byte 4	Byte 3	Byte 2	Byte 1
00000100	01001100	01001111	10110000

Floats & Doubles

- When you need a variable that can hold real numbers (with decimals), you need to either use float or double.
- Floats and Doubles provide a highly accurate estimate of the real number

```
float highTemp;
double distanceMoon;
float goldenRatio;
```

```
goldenRatio = 1.61803; // Compiler really stored: 1.618029952049255371 09375
```

Float Encoding Formula: sign * 2^{exponent} * mantissa

Sign (1 bit)	Exponent (8 bits)	Mantissa (23 bits)
0	01111111	10011110001101110011011

Fun calculator: https://www.h-schmidt.net/FloatConverter/IEEE754.html

Boolean

- Booleans can be either true or false
  - false = 0
  - true = Any other non zero number (by default 1)

```
bool isEnrolled;
bool error;
bool buttonPressed;
```

```
buttonPressed = true;
```

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
							1

Characters

- Character variables store a single character of text.

```
char menuSelect;
char initial;
char tempUnit;
```

- Compiler will convert a letter to an numeric value using an ASCII table

```
tempUnit = 'F';  // Notice the use of single quotes for a single character.
```

```
'F' = ASCII Code 70
```

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0	1	0	0	0	1	1	0

Link to ASCII Table: http://en.cppreference.com/w/cpp/language/ascii

Text

- You can combine characters into a string by using a character array
- You must declare the size of the string and you can't change it.

```
char firstName[256];
char planet[20];
char phoneNumber[13];

         phoneNumber = "555-123-4567";    // Notice you need quotes
```

0	1	2	3	4	5	6	7	8	9	10	11	12
5	5	5	-	1	2	3	-	4	5	6	7	\0

Notice we start at position 0 and not position 1. The character array has 13 positions labeled 0 to 12.

\0 is called the NULL character. The compiler will put this at the end of your character array … make sure you have enough room!

Example

- Write some code in a main function that will declare variables to store the following information:
  - First Name
  - Height (in tenths of an inch)
  - Year born
- Remember to use camelBack rules for writing your variable.
- Select the best data type for each variable.

Input from the Keyboard

- The `cin` command reads data off the input stream from the keyboard.
- Just like `cout`, you can read one or more things from the stream.
- To read something from the input stream, you will need variables declared first.

```
char firstName[20];
int age;

cout << "Enter your first name and age: ";
cin >> firstName >> age;
cout << firstName << " is " << age << " years old." << endl;
```

- The compiler will expect the user to do the following:
  - Put a space between the two inputs
  - Press Enter at the end.

Input from the Keyboard

- The `cin` command will only read one word at a time from the stream.
- To read a whole line from the stream:

```
char address[256];

cout << "Enter your address: ";
cin.getline(address, 256);
cout << "Your address is: " << address << endl;
```

- The `getline` function takes two parameters:
  - The name of character array variable
  - The length of the character array (make sure it matches your declaration).

Example

- In the code you wrote before, prompt the user for information to put in your variables

```
Enter First Name: Chad
Enter Height and Year Born: 69.5 1976
```

Example

- In the code you wrote before, display the following output after getting the information from the user:

```
Chad is 69.5 inches tall.
Chad was born in 1976.
```

Looking Forward

- Saturday End of Day
  - 02 Ponder – Monthly Budget Due at 11:59pm

- Before Class on Monday
  - 1.3 Prepare
    - Read Chapter 1.3 Expressions
    - Submit assign13