CSCI 135
Control Flow (Iteration II)

Subash Shankar

Birthday Example

Spec: Input a person's age, and print "Happy Birthday" centered in a frame of asterisks, where the frame is a 5 by age rectangle. Ex (age=20):

```
********************
*                  *
*  Happy Birthday  *
*                  *
********************
```

1. What are the program's inputs and outputs?

Birthday Example

Spec: Input a person's age, and print "Happy Birthday" centered in a frame of asterisks, where the frame is a 5 by age rectangle. Ex (age=20):

```
********************
*                  *
*  Happy Birthday  *
*                  *
********************
```

1. What are the program's inputs and outputs?
   Age is the only input (an int), and the output is ...
2. What are legal values of age?

Birthday Example

Spec: Input a person's age, and print "Happy Birthday" centered in a frame of asterisks, where the frame is a 5 by age rectangle. Ex (age=20):

```
********************
*                  *
*  Happy Birthday  *
*                  *
********************
```

1. What are the program's inputs and outputs?
   Age is the only input (an int), and the output is ...
2. What are legal values of age?
   Age > 0; moreover, it must be ≥ 16 (since the message has 14 characters). Bad specification - what do we do if $age < 16$? We will choose to print the message without a frame.

Birthday Example - Attempt 1

```
int age;
// input age, ensuring that its positive
do {        // get positive age
   cout << "Enter age (>0) ";
   cin >> age;
} while (age<=0);

// if age < 16, print happy birthday message
if (age < 16)     // No frame printed if age < 16
   cout << "Happy Birthday" << endl;

// if age >15, print happy birthday message in frame
else {                // Print framed message
   What is a frame???
}
```

Birthday Example - Frame Pseudocode

```
int age;
do {          // get positive age
   cout << "Enter age (>0) ";
   cin >> age;
} while (age<=0);
// Print happy birthday message
if (age < 16)     // No frame printed if age < 16
   cout << "Happy Birthday" << endl;
else {                // Print framed message
   // print line 0 of frame: age asterisks
   // print line 1 of frame: * <age-2 spaces> *
   // print centered message
   // print line 3 of frame: * <age-2 spaces> *
   // print line 4 of frame: age asterisks
}
```

? What kind of loops are best for each line?

Birthday Example - Attempt 2

```
do {        // get positive age
   cout << "Enter age (>0) ";
   cin >> age;
} while (age<=0);
// Print happy birthday message
if (age < 16)     // No frame if age < 16
   cout << "Happy Birthday" << endl;
else {            // Print framed message
   // print line 0 of frame: age asterisks
   for (int i=0; i < age; i++) cout << '*';
   cout << endl;
   // print line 1 of frame: * <age-2 spaces> *
   cout << '*';
   for (int i=0; i < age-2; i++) cout << ' ';
   cout << '*' << endl;
   // print line 2 of frame; centered message
   cout << '*';                          how many spaces?
   for (int i=0; i < ??? ; i++) cout << ' ';
   cout << "Happy Birthday"     string (not char)
   for (int i=0; i < ??? ; i++) cout << ' ';
   cout << '*' << endl;
    repeat line 1 code for line 3
    repeat line 0 code for line 4
}
```

Birthday Example - Attempt 2

```
do {        // get positive age
   cout << "Enter age (>0) ";
   cin >> age;
} while (age<=0);
// Print happy birthday message
if (age < 16)     // No frame if age < 16
   cout << "Happy Birthday" << endl;
else {            // Print framed message
   // print line 0 of frame: age asterisks
   for (int i=0; i < age; i++) cout << '*';
   cout << endl;
   // print line 1 of frame: * <age-2 spaces> *
   cout << '*';
   for (int i=0; i < age-2; i++) cout << ' ';
   cout << '*' << endl;
   // print line 2 of frame; centered message
   cout << '*';                          how many spaces?
   for (int i=0; i < ??? ; i++) cout << ' ';
   cout << "Happy Birthday"     string (not char)
   for (int i=0; i < ??? ; i++) cout << ' ';
   cout << '*' << endl;
    repeat line 1 code for line 3
    repeat line 0 code for line 4
}
```

14 characters in "Happy Birthday", 2 more for asterisks ⇒ (age-16)/2 spaces to left and right of message

Birthday Example - Attempt 3

```
do {        // get positive age
   cout << "Enter age (>0) ";
   cin >> age;
} while (age<=0);
// Print happy birthday message
if (age < 16)     // No frame if age < 16
   cout << "Happy Birthday" << endl;
else {            // Print framed message
   // print line 0 of frame: age asterisks
   for (int i=0; i < age; i++) cout << '*';
   cout << endl;
   // print line 1 of frame: * <age-2 spaces> *
   cout << '*';
   for (int i=0; i < age-2; i++) cout << ' ';
   cout << '*' << endl;
   // print line 2 of frame; centered message
   cout << '*';
   for (int i=0; i < (age-16)/2; i++) cout << ' ';
   cout << "Happy Birthday"
   for (int i=0; i < (age-16)/2; i++) cout << ' ';
   cout << '*' << endl;
    repeat line 1 code for line 3
    repeat line 0 code for line 4
}
```

Birthday Example - Attempt 3

```
do {        // get positive age
   cout << "Enter age (>0) ";
   cin >> age;
} while (age<=0);
// Print happy birthday message
if (age < 16)     // No frame if age < 16
   cout << "Happy Birthday" << endl;
else {            // Print framed message
   // print line 0 of frame: age asterisks
   for (int i=0; i < age; i++) cout << '*';
   cout << endl;
   // print line 1 of frame: * <age-2 spaces> *
   cout << '*';
   for (int i=0; i < age-2; i++) cout << ' ';
   cout << '*' << endl;
   // print line 2 of frame; centered message
   cout << '*';
   for (int i=0; i < (age-16)/2; i++) cout << '
   cout << "Happy Birthday"
   for (int i=0; i < (age-16)/2; i++) cout << '
   cout << '*' << endl;
    repeat line 1 code for line 3
    repeat line 0 code for line 4
}
```

Run/test code:

- n<0: ✓
- n=1: ✓
- n=10: ✓
- n=15: ✓
- n=16: ✓
- n=20: ✓
- n=21: ×

```
Enter age (>0) 21
**********************
*                    *
*  Happy Birthday    *
*                    *
**********************
```

Only works for even age (*i.e.*, age%2==0)! Exercise for reader to fix (or refine spec.) Another problem – lots of repeated code (coming soon: how to avoid)

Nested Loops

Recall: The loop body, S, is any statement.
The statement might itself be a loop! (called nested loop).

Spec: Print a w-wide h-high box of asterisks

? What kind of loop?

Nested Loops

Recall: The loop body, S, is any statement.
The statement might itself be a loop! (called nested loop).

Spec: Print a w-wide h-high box of asterisks

? What kind of loop?
! Fixed number of iterations ⇒ `for`

```
for (int row=0; row<h; row++) {
  for (int col=0; col<w; col++) {
    cout << '*';
    };
  cout << endl;
};
```

Nested Loops - Another example

Spec: Print r rows of a diagonal line going southeast with slope -1

```
*
 *
  ...
    *
```

? Loop structure?

Nested Loops - Another example

Spec: Print r rows of a diagonal line going southeast with slope -1

```
    *
     *
       ...
          *
```

? Loop structure?

! 2D fixed size ⇒ nested for loop

```
for (int row=0; row<r; row++) {
  for (int col=0; col<row; col++) {
    cout << ' ';
    };
  cout << '*' << endl;
};
```

inner loop not entered when row=0

Iteration Guidelines and Caveats

- Pick loop construct based on:
  - Do you know how many iterations, or is it based on a condition?
  - When do you want condition evaluated?
  - Now, which maps best onto problem (so that code is readable)?

  (though 3 types have equivalent expressivity)
- Use proper indentation of loop bodies for readability.
- Easy to be off by 1 iteration. Don't forget – there are $n+1$ ints between 0 and n inclusive. Check yourself with good borderline test cases.
- Make sure each iteration makes progress towards loop condition (or you might not terminate!).
- Don't forget the other caveats from earlier!

Exercises

1. Modify the birthday example to work for all ages. Also add error checking to disallow ages that are too large for a 80-wide screen.
2. Input a string s. Output the number of a's or A's in it that appear before the first z.
3. Draw a circle using character graphics (as in the diagonal line example). Hint: Use the results of the previous condition exercise.