Unit 4c

Division and Module Idioms
Unit Objectives

• Apply division and modulo operation to solve specific conversion problems
APPLICATIONS OF DIVISION AND MODULO
**Integer Division and Modulo Operations**

- Recall integer division discards the remainder (fractional portion)
  - Consecutive values map to the *same values*
- Modulo operation yields the remainder of a division of two integers
  - Consecutive values map to *different values*
  - $x \mod m$ will yield numbers in the range $[0$ to $m-1]$
- Example:
Unit Conversion Idiom

• The unit conversion idiom can be used to convert one value to integral number of larger units and some number of remaining items
  – Examples:
    • Ounces to Pounds and ounces
    • Inches to Feet and inches
    • Cents to Quarters, dimes, nickels, pennies

• Approach:
  – Suppose we have \( n \text{ smaller units} \) (e.g. 15 inches) and a conversion factor of \( k \text{ small units} = 1 \text{ large unit} \), (e.g. 12 inches = 1 foot) then...
  – Using \textbf{integer division} \( (n/k) \) yields the integral number of \textbf{larger} units \( (15/12 = 1 \text{ foot}) \)
  – Using \textbf{modulo} \( (n\%k) \) will yield the remaining number of \textbf{smaller} units \( (15 \% 12 = 3 \text{ inches}) \)
Exercise 1: Unit Conversion Idiom Ex. (Making Change)

• Make change (given 0-100 cents) convert to quarters, dimes, pennies
• cpp/var-expr/change
Exercise 2: Unit Conversion

• Suppose a knob or slider generates a number $x$ in the range 0-255
• Use division or modulo to convert $x$ to a new value, $y$, in the range 0-9 proportionally
• $y = \frac{x}{255}$

Each of the 10 bins = ______ small units
Extracting/Isolating Digits Idiom

• To extract or isolate individual digits of a number we can simply divide by the base

• Use modulus (%) to extract the least-significant digits

• Use integer division (/) to extract the most-significant digits
Exercise 3: Isolating Digits Idiom

• Simulate 2 random coin flips producing 2 outcomes (H or T with 50/50 prob.)
• Use `rand()` to generate a random number.
  – `rand()` is defined in `<cstdlib>`
  – Returns a random integer between 0 and \(2^{31}\)
    • Really +2\(^{31}\)-1
  – Your job to convert r1 and r2 to either 0 or 1 (i.e. heads/tails) and save those values in flip1 and flip2

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

int main()
{
    // Generate a random number
    int r1 = rand();
    // And another
    int r2 = rand();
    int flip1 = _____________
    int flip2 = _____________
    cout << flip1 << flip2 << endl;
    return 0;
}

flip1 = ______________
flip2 = ______________
```
Divisibility / Factoring Idiom

- **Modulo** can be used to check if \( n \) is divisible by \( k \)
  - Definition of divisibility is if \( k \) divides \( n \), meaning remainder is 0
- To factor a number we can **divide** \( n \) by any of its divisors

  - \( 12 \ % \ 5 = 2 \)
    - \( \Rightarrow 12 \) is NOT divisible by 5
  - \( 12 \ % \ 3 = 0 \)
    - \( \Rightarrow 12 \) is divisible by 3
  - \( 12 \ / \ 3 = 4 \)
    - \( \Rightarrow 4 \) remains after
    - \( \Rightarrow \) factoring 3 from 12
Challenge Exercise

- **cpp/var-expr/in_n_days**
  - Given the current day of the week (1-7) add \( n \) days and indicate what day of the week (1-7) it will be then

- **Write out table of examples**
  - Input => Desired Output

- **Test any potential solution with some inputs**
  - \( C_{\text{day}} = 1, n = 2 \)...desired outcome = 3
  - \( C_{\text{day}} = 1, n = 6 \)...desired outcome = 7

- **Plug in several values, especially edge cases**

```cpp
int main()
{
    int cday, n;
    cin >> cday >> n;
    int day_plus_n = ______________________;
    return 0;
}
```

<table>
<thead>
<tr>
<th>( n ) (assuming ( c_{\text{day}} = 1 ))</th>
<th>Day_plus_n (desired)</th>
<th>( n ) (assuming ( c_{\text{day}} = 4 ))</th>
<th>Day_plus_n (desired)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>