CSCI 1300

Spring 2022 - Starting Computing

Supriya Naidu and Tom Yeh

Overview

- This Week
- Practicum
- Stats
- Quiz Review
- Q&As
- Textbook
- Project
- Hall of Fame
- Weird but True

This Week

- Recitation: (Wednesday) midnight
- Homework:
    - Start Early: (Wednesday) midnight
    - Final: Due (Saturday) 6pm
- Week 3-2-1: Due (Saturday) midnight
- Quiz: Due (Sunday) 6pm
- Practicum 1: Feb 7!!!

Practicum

- Date: Feb 7, 7:30pm
- Practice
    - 18 students have attempted
- Review sheet
- Review session

What is the output of the following code?

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

1. garbage
2. 33.5
3. value
4. 33

- Answer: value (1/3 got wrong)

Stats

- Enrollment: 534 -> 524 students
- 3-2-1: 463 (87%) -> 411 (78%) -> **373 (71%)** students
- Recitation: 496 (93%) -> 339 (65%) -> **446 (85%)** students
- Homework: 497 (93%) -> 454 (87%) -> **522 (100%)** students!!!!
    - Start early: **448 (85%)**
    - Extra credit: **377 (72%)**
- Quiz: 480 (90%) -> 479 (91.4%) -> **481 (92%)** students
- EdSTEM:
    - 202 (38%) -> 270 (52%) -> **310 (59%)** students
    - 15 -> 39 -> **69** questions

Quiz 3 Review

Compare two strings

```
string some_string1 = "his";
string some_string2 = "cycle";
if (some_string1 < some_string2)
{
    cout << some_string2;
}
else
{
    cout << some_string1;
}
```

1. error
2. hiscycle
3. cycle
4. his

- Answer: 3

`switch` statement

The switch statement in C++

1. is a compound statement that tests all branches against different variables
2. makes the break statement optional
3. is like a sequence of if statements that compares only a single integer value
4. requires compound boolean expressions as alternatives

- Answer 2

3-2-1 Q&As

AnnalieL: Is there ever an advantage to using a switch function?

- My internship at Radical Entertainment
- Program sound effects
- What's wrong?

```
#define KICK 1
#define PUNCH 2
#define HIT 3
...
switch (event) {
    case KICK:
        play("kick.wav");
    case PUNCH:
        play("punch.wav");
    case HIT:
        play("hit.wav");
}
```

- Missing `break;`

Project

Let the player choose a move

```
int move;
cout << "North (W)" << endl;
cout << "West (A)" << endl;
cout << "South (S)" << endl;
cout << "East (D)" << endl;
cout << "Choose your move:"
cin >> move;
```

Let the player keep playing

```
int move;
while (true)
    cout << "North (W)" << endl;
    cout << "West (A)" << endl;
    cout << "South (S)" << endl;
    cout << "East (D)" << endl;
    cout << "Choose your move:"
    cin >> move;
}
```

Move the player around

```
// set the player's location to (0,0)
int x = 0;
int y = 0;
while (true)
    ...
    cin >> move;
    switch (move) {
        case 'W':
            break;
        case 'A':
            break;
        case 'S':
            break;
        case 'D':
            break;
    }
}
```

Update `(x,y)`

- `case 'W':`
  - `y--;`
- `case 'A':`
  - `x--;`
- `case 'S':`
  - `y++;`
- `case 'D':`
  - `x++;`

Textbook

4.1: `while` Loop

self-check 3

```
n = 1;
while (n < 13)
{
   cout << n << " ";
   n = n + 3;
}
```

self-check 4

```java
while (balance < TARGET) {
    if (year > 0) {
        balance = balance + contribution;
    }
    double interest = balance * RATE / 100.0;
    balance = balance + interest;
    year = year + 1;
}
```

4.2: Hand-Tracing

- self-check 1

Hall of Fame

```
while (max_tries > 0
        && block.nNonce < MAX
        && !ShutdownRequested()) {

    if (CheckProofOfWork(block.GetHash(), block.nBits, chainparams.GetConsensus()))
        break;

    ++block.nNonce;
    --max_tries;
}
```

4.4: `do` Loop

self-check 2

```cpp
    do {
     // Keep prompting the user until the input is correct
     cout << "Enter two positive integers, the first smaller than the second."
         << endl;
     cout << "First: " << endl;
     cin >> a;
     cout << "Second: " << endl;
     cin >> b;
    } while (????????);
```

`while (???????)`

> The loop should stop when we got two positive numbers a and b, and a is less than b

- wrong first attempt

  ```
  while (a > 0 && b > 0 && a < b);
  ```

- negate the stopping condition

  > The loop should CONTINUE while it is NOT the case that ....

  ```
  while (!(a > 0 && b > 0 && a < b));
  ```

- apply De Morgan's Law

  ```
  while (!(a > 0) || !(b > 0) || !(a < b));
  ```

- simplify

  ```
  while (a <= 0 || b <= 0 || a >= b);
  ```

Weird But True

- Civilization
- Gandhi
- `int agression_level = 1`
- Democracy and Nuclear Wars
- Fix
    - type
    - condition