

SSE2034: System Software Experiment 3 Spring 2016

Jinkyu Jeong (jinkyu@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
<http://csl.skku.edu>



For Each

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

int main() {
    const int N = 3;
    int scores[N] = {90, 95, 100};

    for (auto n : scores)
        cout << n << endl;
}
```

for-each-element in array:

```
for (element declaration : array)
    statement;
```

For (C-style)

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

int main() {
    const int N = 3;
    int scores[N] = {90, 95, 100};
    int i;

    for (i=0; i<N; i++)
        cout << scores[i] << endl;
}
```

```
for (init_exp; cond_exp; update_exp)
{
    loop_body
}
```

Reference Type

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

int main() {
    const int N = 3;
    int scores[N] = {90, 95, 100};

    for (auto &n : scores)
        cout << n << endl;
}
```

Use reference (pointer) instead of copying element (which may be expensive)

Swap (C-style)

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

void swap (int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}

int main() {
    int x = 3, y = 5;
    swap (&x, &y);
    cout << x << ' ' << y << endl;
}
```

Pass pointers

Swap (C++ style)

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

void swap (int &a, int &b) {
    auto t = a;
    a = b;
    b = t;
}

int main() {
    int x = 3, y = 5;
    swap(x, y);
    cout << x << ' ' << y << endl;
}
```

call-by-reference

call-by-reference

Random Numbers

- Pseudo random number generator

```
#include <iostream>
#include <cstdlib> // c standard lib

using namespace std;

int main() {
    srand(2312); // set initial seed value

    for (int i =0; i<10; i++)
        cout << rand() << endl;
}
```

srand(), rand()

Pseudo random:

Sequence of numbers from rand() is the same, if seed value is the same

Random Numbers (2)

- Set seed from `time()`

```
#include <iostream>
#include <cstdlib> // c standard lib
#include <ctime>   // c time lib
using namespace std;

int main() {
    srand( (time(NULL)) );
    for (int i =0; i<10; i++)
        cout << rand() << endl;
}
```

`time()`

`time(NULL)` returns the number of seconds since 00:00 hours, Jan 1, 1970 UTC

[Lab – Practice #1]

▪ **int randRange(min, max)**

- Command line: number of random numbers
- Input: min, max (check the values if $\text{min} \leq \text{max}$)
- Output: random numbers in the range min to max
inclusive ($\text{min} \leq \text{output} \leq \text{max}$)

```
$ ./randRange 5  
min? 10  
max? 100  
6  
97  
34  
21  
59
```

[Lab – Practice #2]

▪ **Blackjack game**

- `shuffle_cards(num_decks)`
 - One card deck consists of 52 cards (13 numbers x 4 suits)
 - Suits: spade, heart, diamond, clobber
 - Use N decks of cards (N is command-line argument)
- `draw_card()`
 - Return one card in order from deck (`std::vector<string>`)
 - E.g.) 1h (one heart), 13s (king spade), ...
- **House Rule**
 - House wins on draw
 - House always hits below 13 (if `house_sum < 13`)
 - House hits or stands with $\frac{1}{2}$ probability between 13 and 16
 - House always stands on 17 or above (if `house_sum >= 17`)

[Project #1] Blackjack

- **Extend blackjack to multi-player game**
 - Command line option: number of players
 - Total players = house + players
 - Player1 plays in interactive mode

```
Bats: 500
```

```
House      : 3h, X      ( > 3 )
Player (500) : 5s, 10h  (15)
A11 (500)  : 10d, 8c   (18)
A12 (500)  : 2s, 2d    (4)
A13 (500)  : 1d, 4c    (5 or 15)
```

```
Turn of Player --- Choose your action: Hit(H), Stand(S), Split(P)
Action : H
Player : 5s, 10h, 3h (18)
Action : S
```

```
Turn of A11
A11 (500) : 10d, 8c (18)
Action : H
A11 (500) : 10d, 8c, 1c (19)
Action : S
```

Project Rule #1

- **Blackjack Rule에 따라 구현**
 - <http://www.wikihow.com/Sample/Blackjack-Rules>
- **멀티 유저, (사용자를 제외한 플레이어는 AI)**
 - 덱의 크기는 플레이어 수만큼 증가함.
 - 프로그램 시작 시 AI의 개수 및 등급을 지정
 - AI는 카드 카운팅을 한다(LRU)
 - 카드 카운팅 정보를 통해서 확률적으로 카드를 뽑는다.
 - *모르면 검색
 - 등급
 - Dummy – 카드 카운팅 X
 - Low – 전체 덱의 25%
 - Middle – 전체 덱의 50%
 - High – 전체 덱의 75%
 - Master – 전체 덱의 100%
- **UI는 Blackjack_interface.txt에 기반으로 구성**

추가기능

▪ Doubling Down

- 처음 2장의 카드를 받았을 때 배팅액을 2배로 할지 선택한다. 이 경우 추가적으로 1장의 카드를 더 받는 action 혹은 stand하는 action 만 취할 수 있다.

추가기능

▪ Splitting

- 같은 카드를 2장 받았으면 현재 배팅금액을 한 번더 배팅 후 카드를 분리하여 서로 다른 2개의 플레이어 로 운영 가능

Project Submit Rule

- **Deadline : 2016-03-29 6:00PM**
- **Submit**
 - Email: hjjeong@csl.skku.edu
 - Format: [SW3-P1] 학번_이니셜
 - P1_학번_이니셜.zip
 - cpp file
 - ppt file
 - README
 - Makefile
 - gunzip으로 압축 해제 시 P1_학번_이니셜 디렉토리가 생기게끔 압축

Makefile

- **CC = g++**
- **PROG = black_jack**
- **OBJS = main.o**

- **\$(PROG) : \$(OBJS)**
- **\$(CC) -std=c++11 -o \$(PROG) \$(OBJS)**