

# Programming with C++

## COMP2011: Examples on C++ Basics and Controls

Cecia Chan  
Cindy Li

Department of Computer Science & Engineering  
The Hong Kong University of Science and Technology  
Hong Kong SAR, China



# Part I

## Guess The Number



- The game program picks a random number in the range of 1 to 100.
- Two players take turns to guess the number.
- After each guess, the program should tell the player if the number is correct, larger than or smaller than their guessed number.
- Whoever first guesses correctly wins the game.

# Typical Output

Player 1, please enter your guess:

15

Sorry, the number is smaller than 15

Player 2, please enter your guess:

9

Sorry, the number is bigger than 9

Player 1, please enter your guess:

10

Player 1, you win!!!

- Validate that a guessed number is in the range set by the program.
  - request a player to enter again until the input is valid.
- Determine if a guess is correct.
- Give suitable feedback to the players.
- Keep running until a guess is correct.

# First Attempt: 1 Player and 1 Round

```
1  #include <iostream>      /* File: guess1.cpp */
2  using namespace std;
3
4  int main()               // 1st attempt: 1 player, 1 round
5  {
6      int number = 10;     // The answer is fixed beforehand
7      int guess;
8
9      cout << "Player 1, please enter your guess:" << endl;
10     cin >> guess;
11
12     if (guess == number)
13         cout << "Player 1, you win!!!" << endl;
14
15     else if (guess < number)
16         cout << "Sorry, the number is bigger than " << guess << endl;
17
18     else
19         cout << "Sorry, the number is smaller than " << guess << endl;
20
21     return 0;
22 }
```

## Second Attempt: 1 Player and Multiple Rounds

```
1  #include <iostream>      /* File: guess2.cpp */
2  using namespace std;
3
4  int main()               // 2nd attempt: 1 player, multiple rounds
5  {
6      int number = 10;     // The answer is fixed beforehand
7      int guess;
8
9      do                   // Add a loop to implement multiple rounds
10     {
11         cout << "Player 1, please enter your guess:" << endl;
12         cin >> guess;
13
14         if (guess == number)
15             cout << "Player 1, you win!!!" << endl;
16
17         else if (guess < number)
18             cout << "Sorry, the number is bigger than "
19                 << guess << endl;
20         else
21             cout << "Sorry, the number is smaller than "
22                 << guess << endl;
23     } while (guess != number);
24
25     return 0;
26 }
```

# Third Attempt: 1 Player, Multiple Rounds, Fixed Range

```
1  #include <iostream>      /* File: guess3.cpp */
2  using namespace std;
3
4  int main()      // 3rd attempt: 1 player, multiple rounds, fixed range
5  {
6      int number = 10;      // The answer is fixed beforehand
7      int guess;
8      int low = 1, high = 100; // Add 2 variables to record the range
9
10     do // Add a loop to implement multiple rounds
11     {
12         cout << "Player 1, please enter your guess:" << endl;
13         cin >> guess;
14
15         while (guess < low || guess > high) // Input validation loop
16         {
17             cout << "Invalid input, please enter a number between "
18                 << low << " and " << high << endl;
19             cin >> guess;
20         } // Can this loop be replaced with do-while?
21
22         if (guess == number)
23             cout << "Player 1, you win!!!" << endl;
24
25         else if (guess < number)
26         {
27             cout << "Sorry, the number is bigger than "
```



## Third Attempt: 1 Player, Multiple Rounds, Fixed Range ..

```
28         << guess << endl;
29         low = guess + 1; // Update the lower bound of the range
30     }
31     else
32     {
33         cout << "Sorry, the number is smaller than "
34              << guess << endl;
35         high = guess - 1; // Update the upper bound of the range
36     }
37 } while (guess != number);
38
39 return 0;
40 }
```

# Final Code with a Randomly Generated Guess Number

```
1  #include <iostream>          /* File: guess-number.cpp */
2  #include <stdlib.h>          // Needed for calling the rand() function
3  #include <time.h>            // May need for calling the time() function
4  using namespace std;
5
6  int main()    // 2 players, multiple rounds, fixed range, random number
7  {
8      /* Random number generation */
9      srand(time(0));    // Seed the random number generator
10     int number = rand() % 100 + 1;    // Generate a random no. in [1..100]
11
12     int guess;
13     int low = 1, high = 100;
14     int player = 1;    // Set Player 1 as the current player
15
16     cout << "The generated number is: " << number << endl;
17     do
18     {
19         cout << "Player " << player
20              << ", please enter your guess: " << endl;
21         cin >> guess;
22
23         while (guess < low || guess > high) // Input validation loop
24         {
25             cout << "Invalid input, please enter a number between "
26                  << low << " and " << high << endl;
27             cin >> guess;
```

# Final Code with a Randomly Generated Guess Number ..

```
28     }
29
30     if (guess == number)
31         cout << "Player " << player << ", you win!!!" << endl;
32
33     else if (guess < number)
34     {
35         cout << "Sorry, the number is bigger than "
36              << guess << endl;
37         low = guess + 1; // Update the lower bound of the range
38     }
39     else
40     {
41         cout << "Sorry, the number is smaller than "
42              << guess << endl;
43         high = guess - 1; // Update the upper bound of the range
44     }
45
46     player = (player % 2) + 1; // This makes 1 → 2 and 2 → 1
47 } while (guess != number);
48
49 return 0;
50 }
```

## Part II

# Draw an Isosceles Right-Angled Triangles (RATs)



# Draw Triangles

- Design a program that prints some isosceles right-angled triangles (RAT), and allows users to set their size.
- A RAT that has a size of 4 looks like this:

```
*  
**  
***  
****
```

- Furthermore, try the following variations:

Fat RAT	Hollow RAT	Upside-down RAT
<pre>* ** *** **** *****</pre>	<pre>* ** * * * * *****</pre>	<pre>***** ***** *** ** *</pre>

# A Single RAT

```
1  #include <iostream>          /* File: one-rat.cpp */
2  using namespace std;
3
4  int main()
5  {
6      cout << "Size of a RAT: " << endl;
7      int size;                // height = width = size
8      cin >> size;
9
10     for (int width = 1; width <= size; width++) // #iters=height
11     {
12         // Draw one row of a RAT
13         for (int j = 0; j < width ; j++)        // width of a row
14             cout << '*';
15
16         cout << endl;
17     }
18
19     return 0;
20 }
```

# Various RATs

```
1  #include <iostream>          /* File: various-rats.cpp */
2  using namespace std;
3
4  int main()
5  {
6      cout << "Size of a RAT: " << endl;
7      int size;
8      ↪ // height = width = size
9      cin >> size;
10
11     cout << "A fat RAT" << endl;
12     for (int i = 1; i <= size; i++)          // #iterations = height
13     {
14         for (int j = 0; j < i*2 - 1 ; j++) // width of a row
15             cout << '*';
16         cout << endl;
17     }
18
19     cout << "A hollow RAT" << endl;
20     for (int i = 1; i <= size; i++)
21     {
22         for (int j = 0; j < i ; j++)
```

# Various RATs ..

```
22         cout << ((j == 0 || j == i - 1 || i == size) ? '*' :  
23             ↪ ' ');  
24     cout << endl;  
25 }  
26  
27 cout << "An upside-down RAT" << endl;  
28 for (int i = size; i >= 1; i--)  
29 {  
30     for (int j = 0; j < i; j++)  
31         cout << '*';  
32     cout << endl;  
33 }  
34  
35 return 0;  
36 }
```



# A Bug in a RAT: What's Wrong?

```
1  #include <iostream>          /* File: one-bad-rat.cpp */
2  using namespace std;
3
4  int main()
5  {
6      cout << "Size of a RAT: " << endl;
7      int size;
8      cin >> size;
9
10     cout << "A simple RAT:" << endl;
11     for (int i = 0; i < size; i++)
12     {
13         for (int j = 0; j <= i; j++)
14             cout << '*';
15         cout << endl;
16     }
17
18     cout << "Is this a RAT?" << endl;
19     for (int i = 1; i <= size; i++)
20         for (int j = 0; j < i * 2 - 1 ; j++)
21         {
22             cout << '*';
23             cout << endl;
24         }
25
26     return 0;
27 }
```

# A Row of RATs

Now try this:

```
*      *      *      *      *      *      *      *      *      *
**     **     **     **     **     **     **     **     **     **
***    ***    ***    ***    ***    ***    ***    ***    ***    ***
****   ****   ****   ****   ****   ****   ****   ****   ****   ****
***** ***** ***** ***** ***** ***** ***** ***** *****
```

You'll need to measure the width of your screen first.

# Bugs in RATs: What's Wrong?

```
1  #include <iostream>      /* File: row-of-bad-rats1.cpp */
2  using namespace std;
3
4  int main()
5  {
6      cout << "Size of a triangle: " << endl;
7      int size;
8      cin >> size;
9
10     // Find out the number of RATs in a row
11     const int TOTAL_NUM_COLUMNS = 105; // Assumed screen width
12     int num_RATs = TOTAL_NUM_COLUMNS / size;
13
14     for (int i = 1; i <= size; i++)
15     {
16         for (int n = 0; n < num_RATs; n++)
17             for (int j = 0; j < i ; j++)
18                 cout << '*';
19
20         cout << endl;
21     }
22
23     return 0;
24 }
```

# Bugs in RATs Again: What's Wrong?

```
1  #include <iostream>      /* File: row-of-bad-rats2.cpp */
2  using namespace std;
3
4  int main()
5  {
6      cout << "Size of a triangle: " << endl;
7      int size;
8      cin >> size;
9
10     const int TOTAL_NUM_COLUMNS = 105; // Assumed screen width
11     int num_RATs = TOTAL_NUM_COLUMNS / size;
12
13     for (int i = 1; i <= size; i++)
14         for (int n = 0; n < num_RATs; n++)
15             {
16                 for (int j = 0; j < i ; j++)
17                     cout << '*';
18                 for (int j = 0; j < (size-i); j++)
19                     cout << ' '; // Print enough spaces after a RAT
20
21                 cout << endl;
22             }
23
24     return 0;
25 }
```

# A Row of Good RATs

```
1  #include <iostream>                                /* File: row-of-rats.cpp */
2  using namespace std;
3
4  int main()
5  {
6      cout << "Size of a triangle: " << endl;
7      int size;
8      cin >> size;
9
10     // Find out the number of RATs in a row
11     const int TOTAL_NUM_COLUMNS = 105; // Assumed screen width
12     int num_RATs = TOTAL_NUM_COLUMNS / size;
13
14     for (int i = 1; i <= size; i++)
15     {
16         for (int n = 0; n < num_RATs; n++)
17         {
18             for (int j = 0; j < i ; j++)
19                 cout << '*';
20             for (int j = 0; j < (size-i); j++)
21                 cout << ' '; // Print enough spaces after each RAT
22         }
23         cout << endl;
24     }
25
26     return 0;
27 }
```