

C++ Review (+ Practice Quiz)

ROB 102: Introduction to AI & Programming

2021/09/27

Administrative

Wednesday lecture & Friday lab: Project 1 Hacking (due on Monday!)

Quiz 1 in class on Monday, Oct 4th

Quizzes will take place the day of the deadline for each assignment.

There will be a few pop quizzes throughout the semester.

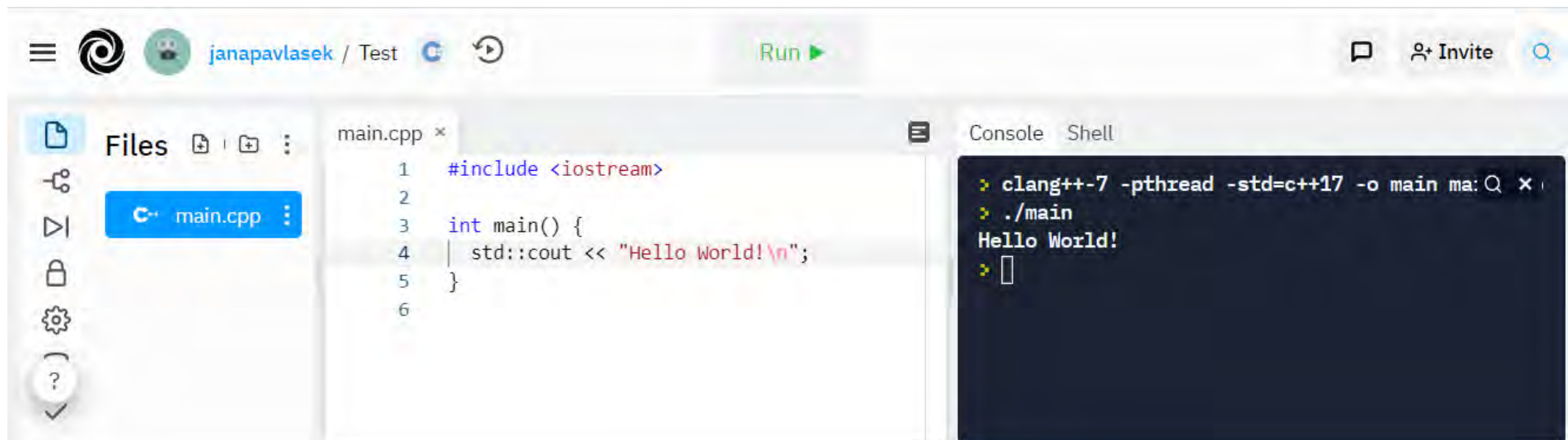
C++ Review

1. Join the Gradescope classroom
2. Complete Quiz 0 individually (15 min)
3. Discuss your solutions in groups
4. Review questions as a class

Repl.it

[Repl.it](https://repl.it) is a website that lets you write and execute code in the browser. It supports many languages including C++!

You can create a free account (or use GitHub) *but code you create is public.*



The screenshot shows the Repl.it web interface. At the top, there is a navigation bar with a menu icon, a profile icon for 'janapavlassek', the text 'Test', a 'Run' button, and an 'Invite' button. Below the navigation bar, there is a file explorer on the left showing a file named 'main.cpp'. The main area is a code editor with the following C++ code:

```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello World!\n";
5 }
6
```

On the right side, there is a terminal window with the following output:

```
> clang++-7 -pthread -std=c++17 -o main ma:Q x |
> ./main
Hello World!
> []
```

Quiz 0: Group Activity

Discuss your answers as a group and come to a consensus on the answers. Then, answer the following questions:

Q1: What is the value of x if the brackets are removed?

Q2: How would you use Function A to get the correct answer?

Q3: How would you modify the code to print the first 10 elements of the Fibonacci sequence?

Q4: How would you modify the code to print "A"?

Q5: How would you modify the code to find the index of the maximum value?

Q1: Operators

What is the value of `x` below?

```
float x = 16 / 4 * (3 + 1);
```

Enter your answer here

Save Answer

Q1: Operators

Try it!

main.cpp ×

```
1  #include <iostream>
2
3  int main() {
4      float x = 16 / 4 * (3 + 1);
5      std::cout << "x = " << x << "\n";
6  }
7
```

Console Shell

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
x = 16
> █
```

Q1: Operators

Recall: Order of operations

PEDMAS = Parentheses, Exponents, Multiplication and Division, Addition and Subtraction

What is the value of `x` below?

```
float x = 16 / 4 * (3 + 1);
```

The diagram illustrates the order of operations for the expression `16 / 4 * (3 + 1)`. It shows three levels of evaluation:

- Level 1: The innermost parentheses `(3 + 1)` are evaluated to `4`.
- Level 2: The division `16 / 4` is evaluated to `4`.
- Level 3: The multiplication `4 * 4` is evaluated to `16`.

Q1: Operators

Recall: Order of operations

PEDMAS = Parentheses, Exponents, Division and Multiplication, Addition and Subtraction

What is the value of `x` below?

```
float x = 16 / 4 * (3 + 1);
```

$$16 / 4 * 3 + 1 = 13$$

Q2: Functions

Which definition of function `f()` would result in the value of `x` being equal to `15` at the end of this code?

```
int x = 5;  
f(x);
```

Function A

```
int f(int x)  
{  
    return x + 10;  
}
```

Function B

```
void f(int x)  
{  
    x += 10;  
}
```

Q2: Functions

```
main.cpp x
1  #include <iostream>
2
3  int f(int x)
4  {
5      return x + 10;
6  }
7
8  int main() {
9      int x = 5;
10     f(x);
11     std::cout << "x = " << x << "\n";
12 }
13
```

```
Console Shell
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
x = 5
> 
```

The correct value is returned, but it isn't assigned to x.

How can we fix it?

Q2: Functions

main.cpp x

```
1  #include <iostream>
2
3  int f(int x)
4  {
5      return x + 10;
6  }
7
8  int main() {
9      int x = 5;
10     x = f(x);
11     std::cout << "x = " << x << "\n";
12 }
13
```

Console Shell

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
x = 15
> █
```

Now it works!

Q2: Functions

Which definition of function `f()` would result in the value of `x` being equal to `15` at the end of this code?

```
int x = 5;  
f(x);
```

Function A

```
int f(int x)  
{  
    return x + 10;  
}
```

Function B

```
void f(int x)  
{  
    x += 10;  
}
```

Q2: Functions

main.cpp x

```
1  #include <iostream>
2
3  void f(int x)
4  {
5      x += 10;
6      std::cout << "in f: x = " << x << "\n";
7  }
8
9  int main() {
10     int x = 5;
11     f(x);
12     std::cout << "x = " << x << "\n";
13 }
14
```

Console Shell

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
in f: x = 15
x = 5
> █
```

Passing x into the function creates a copy of x. The global copy is not modified.

Q2: Functions

Which definition of function `f()` would result in the value of `x` being equal to `15` at the end of this code?

```
int x = 5;  
f(x);
```

Function C

```
int f(int x)  
{  
    x + 10;  
}
```

Function D

```
void f(int& x)  
{  
    x += 10;  
}
```

Q2: Functions

main.cpp x

```
1  #include <iostream>
2
3  void f(int& x)
4  {
5      x += 10;
6  }
7
8  int main() {
9      int x = 5;
10     f(x);
11     std::cout << "x = " << x << "\n";
12 }
13
```

Console Shell

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
x = 15
> □
```

Passing x into the function *as a reference* modifies the original value of x .

Q3: For Loops

What will be printed when the following code is executed?

```
#include <iostream>

int main()
{
    int x1 = 0, x2 = 1;
    for (int i = 0; i < 10; i++)
    {
        std::cout << x1 << " ";
        x2 += x1;
        x1 = x2;
    }

    std::cout << "\n";
}
```

Q3: For Loops

main.cpp x

```
1  #include <iostream>
2
3  int main()
4  {
5      int x1 = 0, x2 = 1;
6      for (int i = 0; i < 10; i++)
7      {
8          std::cout << x1 << " ";
9          x2 += x1;
10         x1 = x2;
11     }
12
13     std::cout << "\n";
14 }
```

Console Shell

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
0 1 2 4 8 16 32 64 128 256
> █
```

How could we print out the first 10 values of the Fibonacci sequence?

Fibonacci sequence: Each number is the sum of the previous two numbers.

Q3: For Loops

```
main.cpp x
1  #include <iostream>
2
3  int main()
4  {
5      int x1 = 0, x2 = 1;
6      for (int i = 0; i < 10; i++)
7      {
8          std::cout << x1 << " ";
9          int tmp = x2;
10         x2 += x1;
11         x1 = tmp;
12     }
13
14     std::cout << "\n";
15 }
```

```
Console Shell
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
0 1 1 2 3 5 8 13 21 34
> 
```

We need to store the old value of x2 before it is updated.

Q4: If Statements

What does this code print?

```
#include <iostream>

int main()
{
    int a = 22 / 7;
    float pi = 3.14159265;
    if (a > 3)
    {
        if (pi > 3)
        {
            std::cout << "A\n";
        }
        else
        {
            std::cout << "B\n";
        }
    }
    else if (a == 3 && pi > 3)
    {
        std::cout << "C\n";
    }
    else
    {
        std::cout << "D\n";
    }
}
```

main.cpp ×

```
1  #include <iostream>
2
3  int main()
4  {
5      int a = 22 / 7;
6      float pi = 3.14159265;
7      std::cout << "a = " << a << "\n";
8      if (a > 3)
9      {
10         if (pi > 3)
11         {
12             std::cout << "A\n";
13         }
14         else
15         {
16             std::cout << "B\n";
17         }
18     }
19     else if (a == 3 && pi > 3)
20     {
21         std::cout << "C\n";
22     }
23     else
24     {
25         std::cout << "D\n";
26     }
27 }
```

Q4: If Statements

Console Shell

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
a = 3
C
> █
```

Since a is an int, it is truncated to 3.

Since pi is a float, it will be higher than int 3.

main.cpp x

```
1  #include <iostream>
2
3  int main()
4  {
5      float a = 22.0 / 7;
6      float pi = 3.14159265;
7      std::cout << "a = " << a << "\n";
8      if (a > 3)
9      {
10         if (pi > 3)
11         {
12             std::cout << "A\n";
13         }
14         else
15         {
16             std::cout << "B\n";
17         }
18     }
19     else if (a == 3 && pi > 3)
20     {
21         std::cout << "C\n";
22     }
23     else
24     {
25         std::cout << "D\n";
26     }
27 }
28
```

Q4: If Statements

Console Shell

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
a = 3.14286
A
> □
```

If a is a float AND either 22 or 7 is a float, then a will be greater than 3.

Q5: Vectors

After the following code executes, what is the value of `v[idx]`?

```
#include <iostream>

int main()
{
    std::vector<float> v = {2.0, 1.83, 6.1, 6.8, -0.83, 5.2};

    int idx = 0;
    for (int i = 1; i < v.size(); i++)
    {
        int val = v[idx];
        int current = v[i];
        if (current > val)
        {
            idx = i;
        }
    }
}
```

Q5: Vectors

```
main.cpp x
1  #include <iostream>
2  #include <vector>
3
4  int main()
5  {
6      std::vector<float> v = {2.0, 1.83, 6.1, 6.8, -0.83, 5.2};
7
8      int idx = 0;
9      for (int i = 1; i < v.size(); i++)
10     {
11         int val = v[idx];
12         int current = v[i];
13         if (current > val)
14         {
15             idx = i;
16         }
17     }
18     std::cout << "v[idx] = " << v[idx] << "\n";
19     std::cout << "idx = " << idx << "\n";
20 }
```

```
Console Shell
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
v[idx] = 6.1
idx = 2
> []
```

Since `val` is an `int`, the values are truncated to {2, 1, 6, 6, 0, 5}. The code finds the index of the first maximum (6) that appears.

How can we find the index of the maximum value?

Q5: Vectors

```
main.cpp ×
1  #include <iostream>
2  #include <vector>
3
4  int main()
5  {
6      std::vector<float> v = {2.0, 1.83, 6.1, 6.8, -0.83, 5.2};
7
8      int idx = 0;
9      for (int i = 1; i < v.size(); i++)
10     {
11         float val = v[idx];
12         float current = v[i];
13         if (current > val)
14         {
15             idx = i;
16         }
17     }
18     std::cout << "v[idx] = " << v[idx] << "\n";
19     std::cout << "idx = " << idx << "\n";
20 }
```

```
Console Shell
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
v[idx] = 6.8
idx = 3
> []
```

Reading the values as floats preserves the part after the decimal.

Q6: Structs

Assume we have a struct defined as follows:

```
struct Course
{
    std::string dept;
    int id;
};
```

Which code will print out ROB 102?

Option A

```
Course c;
c[dept] = "ROB";
c[id] = 102;
std::cout << c[dept] << " " << c[id] << "\n";
```

Option B

```
Course c;
c[0] = "ROB";
c[1] = 102;
std::cout << c[0] << " " << c[1] << "\n";
```

Q6: Structs

Assume we have a struct defined as follows:

```
struct Course
{
    std::string dept;
    int id;
};
```

Which code will print out ROB 102?

Option C

```
Course c;
c.dept = "ROB";
c.id = 102;
std::cout << c.dept << " " << c.id << "\n";
```

Option D

```
Course.dept = "ROB";
Course.id = 102;
std::cout << Course.dept << " " << Course.id << "\n";
```

Q6: Structs

main.cpp x

```
1  #include <iostream>
2
3  struct Course
4  {
5      std::string dept;
6      int id;
7  };
8
9  int main()
10 {
11     Course c;
12     c.dept = "ROB";
13     c.id = 102;
14     std::cout << c.dept << " " << c.id << "\n";
15 }
```

Console Shell

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
ROB 102
> []
```

Q7: Graphs

A finite state machine describes the behavior of a computer program as:

- a graph of pixels
- a graph of nodes
- a graph of pixels and edges
- a graph of nodes and edges