Exercise Session

Week 14

Adel Gavranović
agavranovic@student.ethz.ch

In the initial handout, there was a reference to a PDF called "topics" that was uploaded to the polybox by mistake. Please note, that the topics in that PDF were neither all nor the only topics that are going to be important for the exam, so you should "ignore" it and instead use the "official" list that can be found at the end of the "Master Handout" on the course website

Overview

▸ polybox for session material ▸ mail to TA

Today's Topics

Introduction

- Last exercise session :'(

Comments on last [code]expert Exercises

- When dealing with nodes, try to visualise! Draw the nodes and pointers and ”act out” a function. Abstraction often helps in these cases

New Node or no new Node?

A function is supposed to create a new Node and return a pointer to it. Inside of that function, we see the following code:

```
    Node new_node(value);
    return &new_node;
```

```
Node* create_node(int val)
{
  ...
}
```

New Node or no new Node?

A function is supposed to create a new Node and return a pointer to it. Inside of that function, we see the following code:

```
Node new_node(value);
return &new_node;
```

Question

The program doesn't (always) behave like we want it to. Why?

New Node or no new Node?

A function is supposed to create a new Node and return a pointer to it. Inside of that function, we see the following code:

```
{
    Node new_node(value);
    return &new_node;
}
```

new_n? Gx whato

Question

The program doesn't (always) behave like we want it to. Why?

Answer

The Node was *not* created *dynamically*, so it gets deallocated at the end of the function scope. The pointer to the address that previously was occupied by the Node still exists, but the address might have changed its contents by the next time we try do dereference the pointer, resulting in *undefined behavior*.

New Node or no new Node?

Question

How to rewrite the function to make it work?

New Node or no new Node?

Question

How to rewrite the function to make it work?

```
{
// A: allocate dynamically!
Node* new_node = new Node(value);
return new_node;
}
```

Node(v).

New Node or no new Node?

Question

How to rewrite the function to make it work?

```
// A: allocate dynamically!
Node* new_node = new Node(value);
return new_node;
```

Remember

Dynamically allocated memory (with `new`) will *not* get deleted automatically at the end of a scope, but *normally* allocated memory will be deleted at the end of a scope.

Questions or Comments re: Exercises?

Learning Objectives Checklist

Now I...

- ☐ understand the concept of subclasses
- ☐ understand the difference between virtual and non-virtual methods
- ☐ can implement simple class hierarchies

Questions?

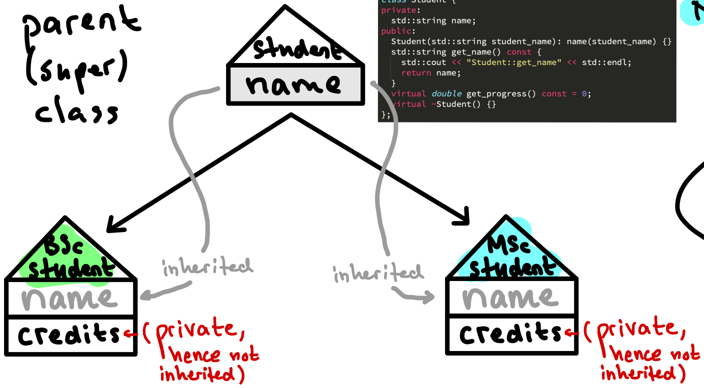
Inheritance and Polymorphism

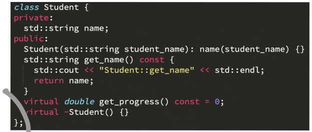
not relevant for the exam, *sooo...*

Inheritance and Polymorphism

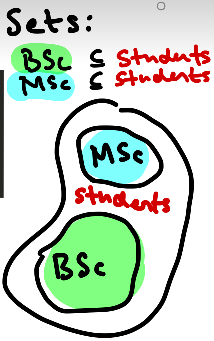
not relevant for the exam, *sooo...*
only a quick overview of the concept

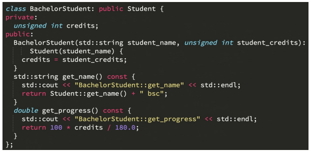
Inheritance and Polymorphism



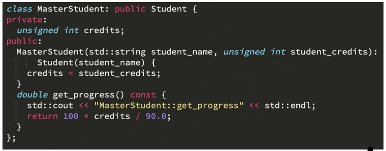


```cpp
class Student {
private:
  std::string name;
public:
  Student(std::string student_name): name(student_name) {}
  std::string get_name() const {
    std::cout << "Student::get_name" << std::endl;
    return name;
  }
  virtual double get_progress() const = 0;
  virtual ~Student() {}
};
```





```cpp
class BachelorStudent: public Student {
private:
  unsigned int credits;
public:
  BachelorStudent(std::string student_name, unsigned int student_credits):
      Student(student_name) {
    credits = student_credits;
  }
  std::string get_name() const {
    std::cout << "BachelorStudent::get_name" << std::endl;
    return Student::get_name() + " bsc";
  }
  double get_progress() const {
    std::cout << "BachelorStudent::get_progress" << std::endl;
    return 100 * credits / 180.0;
  }
};
```



```cpp
class MasterStudent: public Student {
private:
  unsigned int credits;
public:
  MasterStudent(std::string student_name, unsigned int student_credits):
      Student(student_name) {
    credits = student_credits;
  }
  double get_progress() const {
    std::cout << "MasterStudent::get_progress" << std::endl;
    return 100 * credits / 90.0;
  }
};
```

→ now, wherever a 'Student' is a parameter/input, we can pass a BSc- or MSc-Student.

Questions?

"iS tHiS ReLeVaNt fOr ThE eXaM?"

- Everything in the lectures that wasn't specifically marked "not exam relevant" **is** relevant for the exam
- All of the lecture handouts are saved in ▸ one giant handout
- `ctrl+F` if you're not sure if something *could* come up
- Print out the last few pages in the Master-Handout and use them as checklists

”iS tHiS ReLeVaNt fOr ThE eXaM?”

In the initial handout, there was a reference to a PDF called "topics" that was uploaded to the polybox by mistake. Please note, that the topics in that PDF were neither all nor the only topics that are going to be important for the exam, so you should "ignore" it and instead use the "official" list that can be found at the end of the "Master Handout" on the course website

- Everything in the lectures that wasn't specifically marked ”not exam relevant” **is** relevant for the exam
- All of the lecture handouts are saved in ▸ one giant handout
- `ctrl+F` if you're not sure if something *could* come up
- Print out the last few pages in the Master-Handout and use them as checklists
- Things that are on the slides, but weren't talked about much in the lectures *might* still come up if they are close to something that was discussed in class (e.g. $010 = 8_{10}$)

e

↳ octal

"octal"

Last Points re: Exam

- You have to build up an actual *skill* and not just ”know” things: **practice is key**

Last Points re: Exam

- You have to build up an actual *skill* and not just "know" things: **practice is key**
- Make sure you practice on the same keyboard layout as the one that you will have in the exam (CH-DE/US) (changing keyboard layout will *not* be an option)
- @MacUsers: There will be a "cheat sheet" at the exam, but make sure to practice beforehand still

Last Points re: Exam

- You have to build up an actual *skill* and not just ”know” things: **practice is key**
- Make sure you practice on the same keyboard layout as the one that you will have in the exam (CH-DE/US) (changing keyboard layout will *not* be an option)
- @MacUsers: There will be a ”cheat sheet” at the exam, but make sure to practice beforehand still
- **Don’t** use anything that was not covered in class, especially libraries and data structures (stacks, heaps, queues, cmath, ...). Seriously, it might cost you grades

data structures

Last Points re: Exam

- You have to build up an actual *skill* and not just ”know” things: **practice is key**
- Make sure you practice on the same keyboard layout as the one that you will have in the exam (CH-DE/US) (changing keyboard layout will *not* be an option)
- @MacUsers: There will be a ”cheat sheet” at the exam, but make sure to practice beforehand still
- **Don’t** use anything that was not covered in class, especially libraries and data structures (stacks, heaps, queues, cmath, **...**). Seriously, it might cost you grades
- No need for good documentation (if not specifically asked for) but feel free to do so, same goes for `const` (make sure you understand `const` (member)functions!)

Last Points re: Exam

- You can ask me questions during the lernphase. If it's something complicated we might do a zoom call
- Do the mock exam(s?) and the older programming exam tasks
- Don't forget, that you have other exams too, so don't spend *all* of your time on Informatik
- *IMO*, practicing solving problems quickly and with little prepp can help you stay calmer during the exam

Last Points re: Exam

- You can ask me questions during the lernphase. If it's something complicated we might do a zoom call
- Do the mock exam(s?) and the older programming exam tasks
- Don't forget, that you have other exams too, so don't spend *all* of your time on Informatik
- *IMO*, practicing solving problems quickly and with little prepp can help you stay calmer during the exam
- Bring something to write/sketch on. It makes some of the questions so easy it almost feels like cheating

Last Points re: Exam

- You can ask me questions during the lernphase. If it's something complicated we might do a zoom call
- Do the mock exam(s?) and the older programming exam tasks
- Don't forget, that you have other exams too, so don't spend *all* of your time on Informatik
- *IMO*, practicing solving problems quickly and with little prepp can help you stay calmer during the exam
- Bring something to write/sketch on. It makes some of the questions so easy it almost feels like cheating
- Try to be excited, not panicky and don't overdose on caffeine before the exam

Recursion

Here are two really cool videos on the topic of recursion. Watch them once or twice and see if the concept starts to make sense

- ▸ Recursion in general ← "leap of faith" of recursion
- ▸ Towers of Hanoi

Questions?

Q&A

void foo $\underbrace{\text{(const int\& n)}}_{\text{const (no change to input)}}$ vs. $\underbrace{\text{void goo (int\& n) const}}_{\text{function doesn't change anything}}$

set

e.g: print, read, get

```
struct vec2{
private:
   int x,y;
public:
                        const
   void set(const int &x, ... y){
   void print() const;
```

Q&A EBNF

parsing: going through input and saving it to use later.

✓ ✓ ✓ ✓ ✓ ✓

I have a dog.

Task

Rewrite the BNF from the previous slides into an EBNF with the follwing additional syntax:

- `{...}`: at the location of this syntax, the content between the brackets can be repeatet $n \in \{\mathbb{N}_0\}$ times 0, 1, 2 ..
- `[...]`: at the location of this syntax, the content between the brackets can be repeatet $m \in \{0, 1\}$ times

Q&A EBNF

exercise/task " is word/combo valid?"

'set of rules'

{ } => while

```
seq    = term ['_' seq]
term   = 'A' {'a'} | 'a' {'a'}
```

Inputs (Program)

~~A~~aaAa

~~aaaBa~~

~~Baa~~

~~A~~aa

char

```
bool seq(std::istream& is){
    bool valid = term(is);

bool term(std::istream& is){
    bool valid = false;

    valid = (consume(is,'A') || consume(is,'a'))
    while(consume(is,'a')){
        valid = valid &&
```

Q&A

Q&A

Info
1
2
. . .
13
1.1
90
1.2
80
100
1
90%
2
50%
LinAlg
DiskMath
→ Ex. 13.2

What's an if()-chain?

```
if(grid != nullptr){
   if(grid -> is_filled(row, col)){
      if(col == 8){
         if(fillValidNumber(grid, row + 1, 0)){
            return true;
         }
      } else {
         if(fillValidNumber(grid, row, col + 1)){
            return true;
      }
   }
}
```

What's an if()-chain?

```
if(grid != nullptr){
   if(grid -> is_filled(row, col)){
      if(col == 8){
         if(fillValidNumber(grid, row + 1, 0)){
            return true;
         }
      } else {
         if(fillValidNumber(grid, row, col + 1)){
            return true;
         }
      }
   }
}
```

Task

Simplify this mess

Simplifying...

```
if(grid != nullptr){
   if(grid -> is_filled(row, col)){
      if(col == 8){
         return fillValidNumber(grid, row + 1, 0);
      } else {
         return fillValidNumber(grid, row, col + 1);
      }
   }
}
```

Simplifying...

```
if(grid != nullptr){
    if(grid -> is_filled(row, col)){
        if(col == 8){
            return fillValidNumber(grid, row + 1, 0);
        } else {
            return fillValidNumber(grid, row, col + 1);
        }
    }
}
```

Task

Simplify this even further

Simplifying further...

```
if(grid != nullptr && grid -> is_filled(row, col)){
    if(col == 8){
        return fillValidNumber(grid, row + 1, 0);
    } else {
        return fillValidNumber(grid, row, col + 1);
  }
}
```

Simplifying further...

```
if(grid != nullptr && grid -> is_filled(row, col)){
      if(col == 8){
         return fillValidNumber(grid, row + 1, 0);
      } else {
         return fillValidNumber(grid, row, col + 1);
   }
}
```

Trick

Two `if()` can sometimes be put together into a one boolean expression with `&&`

Remember

`&&` *shortcircuits*, so it won't check the second one if the first one returns true

Questions?

THANK YOU!

Best of luck for the exams and see you next semester!