Institut für Informatik — TUM
Scientific **C**omputing in **C**omputer **S**cience
R. Wittmann

WT 2015

Tutorial
(Advanced Programming)
Worksheet 7: Repetition / Exam training

Assignment 1: Program interpretation

Use the table below to track each variable's value during execution of the program to the left of the table. The row for line number l shall contain each variable's value *after* line l of the program has been executed. An x at the top of a table column binds that column to the variable x of the program. $*y$ refers to the *value* pointed to by pointer y. If in any line of your program a variable has not been declared yet, mark it with "–". If its value is undefined, mark it with "u".

```
 1 static int *p;
 2 int func(
 3     int &arg
 4 );
 5
 6 int main() {
 7     int a = 0, b=3,
           c;
 8     double d;
 9     a = ((b&5) - 2);
10     d = func(a);
11     c = (--a)*6/ b;
12     return c + 1;
13 }
14 int func(
15     int &arg
16 ) {
17     p = &arg;
18     return (*p)++;
19 }
```

	variable names				
line number	a	b	c	d	*p
7					
8					
9					
17					
18					
10					
11					

What is the return value of the program?

Assignment 2: Type casts

a) The compiler introduces an implicit type cast for the three assignments in lines 4–6 of the following code snippet. Assume variables `f` and `u` are initialized in a separate context such that the compiler can make no assumption about their contents.

```
1 float f;
2 unsigned int u;
3 ...
4 double d = f;
5 int i = f;
6 int j = u;
```

For each of the casts, explain whether it is unsafe, i.e. whether accuracy is lost or unintended behavior might be caused (1 sentence each).

b) Consider the following code fragment:

```
1  class Base {
2  public:
3   void echo() { std::cout << "Base" << std::endl; };
4  };
5
6  class Derived : public Base {
7  public:
8   void echo() { std::cout << "Derived" << std::endl; };
9  };
10
11 int main() {
12    Base* ptr = new Derived();
13    ptr->echo();
14 }
```

Determine the program's output and give an explanation for the wrong output (≤ 2 sentences).

Further, show two different ways how one could restore the desired output.

Assignment 3: Memory management

There exist various ways to allocate arrays in C/C++. Consider the code snippet below.

```
1 #include <cstdlib>
2 class Pi {
3   public:
4     Pi(): v(new double){*v=3.1415;}
5     ~Pi(){}
6     double get() const {return *v;}
7   private:
8     double *v;
9 };
10
11 int main(){
12   Pi a[3];
13   Pi *b = new Pi[3];
14   Pi *c = (Pi*) malloc(sizeof(Pi)*3);
15
16   // TODO: remove potential memory leaks
17
18   return 0;
19 }
```

1. What is the difference between the three allocation methods (line 12-14)?
2. Detail each allocation and point out the differences between them (total: ca. 3-4 sentences).
3. Extend the function main() such that all potential memory leaks are removed.