

Name: Muhammad Asim Ali  
 Student ID: V00854120  
 Email: maali@uvic.ca  
 Course: SENG475  
 Section: T01

Assignment ID: cpp\_basics  
 Assignment Title: C++ Basics

Submission Source: cpp\_basics-MAsimSENG/  
 Commit ID: ?

#### Submitted Files

=====

```
drwxrwxr-x    4096 2020-05-21 22:32 ./app
-rw-rw-r--      0 2020-05-21 22:32 ./app/test_random.cpp
-rw-rw-r--    367 2020-05-21 22:32 ./CMakeLists.txt
-rw-rw-r--    140 2020-05-21 22:32 ./IDENTIFICATION.txt
drwxrwxr-x    4096 2020-05-21 22:32 ./include
drwxrwxr-x    4096 2020-05-21 22:32 ./include/ra
-rw-rw-r--   1253 2020-05-21 22:32 ./include/ra/random.hpp
drwxrwxr-x    4096 2020-05-21 22:32 ./lib
-rw-rw-r--   2079 2020-05-21 22:32 ./lib/random.cpp
```

#### Results

=====

Package	Operation	Target	Status
nonprog	generate	---	FAIL (1 0.0s 1L)
random_orig	generate	---	OK (0.1s)
random_orig	configure	---	OK (0.8s)
random_orig	build	test_random	FAIL (2 0.1s 2L)
random_sane	generate	---	OK (0.2s)
random_sane	configure	---	OK (0.7s)
random_sane	build	test_random	FAIL (2 0.1s 76L)
rational_orig	generate	---	FAIL (1 0.2s 1L)
rational_sane	generate	---	FAIL (1 0.1s 1L)

Normally, an operation is indicated as having a status of either "OK" or "FAIL". A status of "?" indicates that the operation could not be performed for some reason (e.g., due to an earlier error or being a manual step). The time (in seconds) required for an operation is denoted by an expression consisting of a number followed by the letter "s" (e.g., "5.0s"). In the case of a test that consists of multiple test cases, the number of failed test cases and total number of test cases is expressed as a fraction (e.g., "10/50" means 10 test cases failed out of 50 test cases in total). The length (in lines) of the log file generated by an operation is denoted by an expression consisting of a number followed by the letter "L" (e.g., "10L"). To ascertain the reason for the failure of an operation, check the contents of the log file provided.

#### Legend

=====

Package: nonprog  
 Nonprogramming exercises

Package: random\_orig

The code as originally submitted by the student.

Build target: test\_random

Build the test\_random program.

Package: random\_sane

Code with modifications to perform API sanity checking.

Build target: test\_random

Build the test\_random program.

Package: rational\_orig

The code as originally submitted by the student.

Package: rational\_sane

Code with modifications to perform API sanity checking.

```
1  ERROR: missing file/directory README.pdf
```

```
1  gmake: *** No rule to make target `test_random'.  Stop.  
2  ERROR: build failed to generate executable test_random
```

```
1 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
2 -S/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-rando
3 m_sane/source
4 -B/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-rando
5 m_sane/derived --check-build-system CMakeFiles/Makefile.cmake 0
6 /usr/bin/gmake -f CMakeFiles/Makefile2 test_random
7 gmake[1]: Entering directory
8 `/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random
9 _sane/derived'
10 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
11 -S/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-rando
12 m_sane/source
13 -B/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-rando
14 m_sane/derived --check-build-system CMakeFiles/Makefile.cmake 0
15 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
16 cmake_progress_start
17 /tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random_
18 sane/derived/CMakeFiles 4
19 /usr/bin/gmake -f CMakeFiles/Makefile2 CMakeFiles/test_random.dir/all
20 gmake[2]: Entering directory
21 `/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random
22 _sane/derived'
23 /usr/bin/gmake -f CMakeFiles/ra.dir/build.make CMakeFiles/ra.dir/depend
24 gmake[3]: Entering directory
25 `/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random
26 _sane/derived'
27 cd
28 /tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random_
29 sane/derived &&
30 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
31 cmake_depends "Unix Makefiles"
32 /tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random_
33 sane/source
34 /tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random_
35 sane/source
36 /tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random_
37 sane/derived
38 /tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random_
39 sane/derived
40 /tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random_
41 sane/derived/CMakeFiles/ra.dir/DependInfo.cmake --color=
42 Scanning dependencies of target ra
43 gmake[3]: Leaving directory
44 `/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random
45 _sane/derived'
46 /usr/bin/gmake -f CMakeFiles/ra.dir/build.make CMakeFiles/ra.dir/build
47 gmake[3]: Entering directory
48 `/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random
49 _sane/derived'
50 [ 25%] Building CXX object CMakeFiles/ra.dir/lib/random.cpp.o
51 /home/frodo/public/ugls_lab-4.0.70/bin/c++
52 -I/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-rando
53 m_sane/source/include -pedantic-errors -std=gnu++17 -o
54 CMakeFiles/ra.dir/lib/random.cpp.o -c
55 /tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random_
56 sane/source/lib/random.cpp
57 /tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random_
58 sane/source/lib/random.cpp:2:10: fatal error: random.hpp: No such file or
59 directory
60     2 | #include "random.hpp"
61       |           ^~~~~~
62 compilation terminated.
```

```
63 gmake[3]: *** [CMakeFiles/ra.dir/lib/random.cpp.o] Error 1
64 gmake[3]: Leaving directory
65 `/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random
66 _sane/derived'
67 gmake[2]: *** [CMakeFiles/ra.dir/all] Error 2
68 gmake[2]: Leaving directory
69 `/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random
70 _sane/derived'
71 gmake[1]: *** [CMakeFiles/test_random.dir/rule] Error 2
72 gmake[1]: Leaving directory
73 `/tmp/assignment_precheck-maali@ugls16.ece.uvic.ca-12426-wVHKH2xJ/package-random
74 _sane/derived'
75 gmake: *** [test_random] Error 2
76 ERROR: build failed to generate executable test_random
```

```
1  ERROR: missing file/directory include/ra/rational.hpp
```

```
1  ERROR: missing file/directory include/ra/rational.hpp
```



```
1
2
3  cmake_minimum_required(VERSION 3.1 FATAL_ERROR )
4
5  project(LCG LANGUAGES CXX)
6
7  set(random_headers include/ra/random.hpp)
8  set(random_sources lib/random.cpp)
9
10 add_library(random ${random_sources} ${random_headers})
11
12 target_include_directories(random PUBLIC include/ra/)
13
14 install(TARGETS random DESTINATION lib)
15
16 install(FILES ${random_headers} DESTINATION include/ra)
```

```
1  class linear_congruential_generator {
2
3      // part a : unsigned integer type atleast 64bits
4
5  public:
6      typedef unsigned long long int int_type;
7
8
9      linear_congruential_generator(int_type a, int_type c, int_type m, int_type s
10 );
11      // no need to provide destructor, it will be provided by default by the comp
12      iler.
13      linear_congruential_generator(linear_congruential_generator && L); // move c
14      onstructor T(T&&t)
15      linear_congruential_generator(const linear_congruential_generator &L); // co
16      py constructor : T(const T &t)
17
18      linear_congruential_generator& operator=(const linear_congruential_generator
19      & t); // copy assignment operator definition
20
21      linear_congruential_generator& operator=(linear_congruential_generator&& t);
22      // move assignment operator definiton
23
24      bool operator==(const linear_congruential_generator& L1);
25
26      bool operator!=(const linear_congruential_generator& L1);
27
28      int_type operator() (); // overloading function call
29
30
31      // member function declarations
32
33      int_type multiplier();
34
35      int_type increment();
36
37      int_type modulus();
38
39      static int_type default_seed();
40
41      void seed(int_type s);
42
43      void discard(int_type n);
44
45      int_type min();
46      int_type max();
47
48
49      private:
50          int_type a;
51          int_type c;
52          int_type m;
53          int_type xo;
54          int_type current_state;
55
56
57 };
```

```
1  #include "random.hpp"
2  #include <iostream>
3
4  typedef unsigned long long int int_type;
5
6  linear_congruential_generator::linear_congruential_generator(int_type a, int_type c, int_type m, int_type s = default_seed()) {
7      s=default_seed();
8      this->a =a;
9      this->c = c;
10     this->m = m;
11     if(c % m ==0 && s % m ==0) {
12
13         xo= 1;
14     }
15     else {
16         xo=s;
17     }
18
19     this->current_state = xo;
20 }
21
22 int_type linear_congruential_generator::multiplier() {
23     return a;
24 }
25
26 int_type linear_congruential_generator::increment() {
27     return c;
28 }
29
30 int_type linear_congruential_generator::modulus() {
31     return m;
32 }
33
34 int_type linear_congruential_generator::default_seed() {
35     return 1;
36 }
37
38 void linear_congruential_generator::seed(int_type s) {
39
40     if(c%m ==0 and s%m ==0) {
41
42         xo=1;
43     }
44     else {
45         xo = s;
46     }
47
48 }
49
50 bool linear_congruential_generator::operator==(const linear_congruential_generator& L1) {
51     return (L1.a == this->a && L1.c == this->c && L1.m == this->m && L1.current_state == this->current_state);
52 }
```

```
60
61
62
63 }
64
65 bool linear_congruential_generator::operator!=(const linear_congruential_genera
tor& L1) {
66
67     return !(L1.a == this->a && L1.c == this->c && L1.m == this->m && L1.current_s
tate == this->current_state);
68
69
70
71 }
72
73 int_type linear_congruential_generator::operator()() {
74
75     // return the value corresponding to the next position of generator
76
77     current_state = (a*xo +c) % m;
78
79     return current_state;
80
81
82 }
83
84
85 void linear_congruential_generator::discard(int_type n ) {
86
87     // discards the next n numbers in the generated seq
88
89     // logic just compute the seed over the next n times
90     // those n seeds will not be used and we will advance
91
92     for(int i=0; i<n; i++){
93
94         xo = (a*xo +c) % m;
95
96     }
97
98
99 }
100
101 int_type linear_congruential_generator::min() {
102     // min is 1 if c is 0 otherwise it is 0.
103     return c == 0 ? 1 :0;
104 }
105
106
107 int_type linear_congruential_generator::max() {
108     return m-1;
109 }
110
111 linear_congruential_generator&linear_congruential_generator::operator=(linear_co
ngruential_generator&& L )=default;
112
113 linear_congruential_generator&linear_congruential_generator::operator=(const lin
ear_congruential_generator& L )=default;
```

