UNIVERSITY OF GLASGOW

DEGREE OF MASTER OF ENGINEERING
DEGREE OF BACHELOR OF ENGINEERING
DEGREE OF BACHELOR OF SCIENCE IN ENGINEERING

ADVANCED PROGRAMMING AND SOFTWARE ENGINEERING 3
(ENG3091)

Mock Paper
Time allowed: Two Hours

Answer ALL questions.
The total number of marks on this paper is 100.

The numbers in square brackets in the right-hand margin indicate the marks allotted to the part of the question against which the mark is shown. These marks are for guidance only.

An electronic calculator may be used provided that it does not have a facility for either textual storage or display, or for graphical display.
If a calculator is used, intermediate steps in the calculation should be indicated.

The Command Line

Q1. (a) How does a C++ program indicate success or failure to the calling shell? [1]

(b) The shell uses 0 to represent true and non-zero to represent false, which is the opposite of C++. Why did the authors choose to adopt this convention? [1]

(c) **git**, a popular revision control system, allows one to commit changes made to files locally with the command `git commit`. One might see the following written in shell scripts designed to automate this process:

```
git commit || echo commit failed.
```

Explain what this command achieves and how. [2]

(d) The command **date** prints the current date and time. Write a script using the shell's **if** command which commits changes to the **git** repository and sets a variable **LASTCOMMIT** to the current date and time if the commit succeded.[1] [2]

(e) The **test** commend, which can also be invoked as [, supports the opreation **-nt** ("newer than") to compare the time of two files and **-f** to determine whether a file exists. Explain the operation and purpose of the following shell script:

```
SMALLPHOTOSDIR=../Small
find . -name \*.jpeg | while read photo ; do
  if [ ! -f .timestamp ] || [ "$photo" -nt .timestamp ] ; then
    convert "$photo" -size 320x240 "$SMALLPHOTOSDIR/$photo"
  fi
done
touch .timestamp
```

[12]

(f) Why did the programmer choose the name **.timestamp** instead of **timestamp** for that file? [1]

Question continues on the next page.

[1]Here and elsewhere, the conscientious examiner should not normally be setting memory tests. Look at the rest of the question or paper. You'll probably find similar commands to remind you of the correct syntax if you can't recall it. In the real exam, the material in the Bash "cheat sheet" on the moodle will also be made available to you.

(g) You have exactly the following resources. You change into the Holiday_Photos directory and run the script from Q1(e). State *two* reasons why it will fail.

[4]

```
$ tree --charset=ascii Holiday_Photos
Holiday_Photos
|-- 0001.jpeg
|-- 0002.jpeg
|-- 0005.jpeg
`-- Pool
    |-- 0003.jpeg
    `-- 0004.jpeg

1 directory, 5 files
```

(h) Explain[2] how you would change the script so it would work as expected.

[2]

[2]Writing programs in an exam where you have limited reference materials isn't a good test. But since this is a drill, how about coming back here after you've done the paper and actually writing that code?

Building C++ Programs

Q2. Among many other modules, a chemical plant control program contains a water_temperature module and a serial_number module. The following is the API for the water_temperature module which is to have its source code in `water_temperature.cxx`:

```
#ifndef _WATER_TEMPERATURE_H_

/** Test whether the temperature is in range */
class Temp_Range {
public:
        /** Constructor
         * @param min Minumum permitted temperature in Kelvin
         * @param max Maximum permitted temperature in Kelvin */
         Temp_Range(float min, float max);

         /** Get the status of the system
          * @param temperature The current temperature in Kelvin
          * @returns 0 if within range, +1 if over, -1 if under temp */
          int status(double temperature);

private:
         float max, min; ///< maximum and minumum permissible temperatures
};
#endif
```

and the following is the test suite for the serial_number module with source in `serial_number.cxx`:

```
// File: test_serial_number.cxx

#include "serial_number.h"

int main ()
{
        Serial_Number sn(20); // set initial serial number to 20.

        sn(); sn();
        if (sn() != 22)
                return 1; // fail if serial number isn't incrementing
        return 0; // otherwise return success.
}
```

Question continues on the next page.

(a) Write the test code for the `Temp_Range` class. Observe the coding conventions in the test code used in the question. [3]

(b) Write `serial_number.h`. Observe the coding conventions in the header file used in the question. [8]

(c) Describe the components of a **make** rule. [2]

(d) Write a **Makefile** with a target **tests** which compiles and runs both test programs. Each program should be a separate sub-target. Use **make** variables liberally to ensure your Makefile is easily understood and maintained. You do not need to include the rules for auto-dependency generation. [12]

C++

Q3. A program contains the following code. Line numbers have been added to make it easier for you to refer to them in your answer, but they are not part of the program.

```
01      #include <iostream>
02      #include <memory>

03      void print_inc(std::shared_ptr<int> s)
04      {
05        std::cout << ++*s << std::endl;
06      }

07      int main() {

08        print_inc(std::shared_ptr<int>(new int {0}));

09        std::shared_ptr<int> sip(new int {0});
10        for (int i {0} ; i < 10 ; i++)
11          print_inc(sip);
12      }
```

(a) Write down the type of `print_inc` using English words. [2]

(b) What does the program print out? [2]

(c) What is a **shared_ptr**? Explain how it manages the memory it points at. [8]

(d) In what way is a **unique_ptr** different from a **shared_ptr**? [4]

(e) When `print_inc` is called from line 11, what is the use count of its parameter `s`? Give a reason for your answer. [1]

(f) When `print_inc` is called from line 8, what is the use count of `s`? Again, give a reason for your answer. [1]

(g) A developer replaces all instances of `std::shared_ptr` with `std::unique_ptr`. The program now fails to compile. What simple change can be made to `print_inc`'s argument to fix the problem? Explain why your solution works. [3]

(h) Is the use of **unique_ptr**s to be preferred over

i. **shared_ptr**s, or

ii. primitive C or C++ pointers?

Give reasons. [4]

Software Engineering

Q4. An entry system operates as follows. Initially the door is locked and a red LED is illuminated. A user approaches and places a finger tip on a sensor. If authorised, the red LED is turned off and a green LED illuminated for 5 seconds, during which time the door can be opened. If the user is unauthorised, a blue flashing light and a siren is turned on for 20 seconds.

(a) Identify the states of the system. [3]

(b) For each state identify its entry and exit actions. [8]

(c) Draw a UML state-transition diagram combining all the information from the previous two parts of this question with the transition conditions. [8]

(d) When you code up the system, what C++ language features would you use to correspond to the state, action and transition? [6]