UNIVERSITY OF JOHANNESBURG
FACULTY OF SCIENCE

COMPUTER SCIENCE	**APK CAMPUS**

CSC01B1
INTRODUCTION TO DATA STRUCTURES (C++)

SEMESTER TEST 2

9 OCTOBER 2020

EXAMINER	**Prof. DA Coulter**
MODERATOR	**Prof. DT van der Haar**

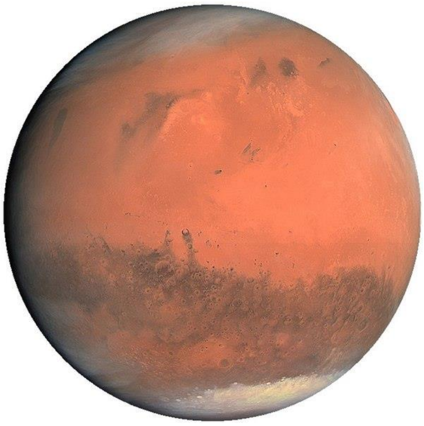
TIME	**3 Hours**	**MARKS**	**100**

Please read the following instructions carefully

1. **You must complete this test yourself within the prescribed time limits.**
2. **You are bound by all university regulations please special note of those regarding assessment, plagiarism, and ethical conduct.**
3. **You may not directly take any code from any source including your own previous submissions. All code must be written by yourself during this the test.**
4. **You must complete and submit the** "`Honesty Declaration : Online Assessment`" **document along with your submission to EVE. No submissions without an accompanying declaration will be marked.**
5. **You may submit scanned pages as per the instructions on EVE**
6. **Your answers to the question (in a single PDF format) together with the declaration must be submitted in a zip archive named in the following format.**
   ```
   STUDENTNUMBER_SURNAME_INITIALS_SUBJECTCODE_ASSESSMENT
   e.g. 202012345_COULTER_DA_CSC01B1_ST2.zip
   ```
7. **Additional time for submission is allowed for as per the posted deadlines on EVE.**
8. **No communication concerning this test is permissible during the assessment session except with Academy staff members.**

Note: Although there are 120 marks available the maximum mark which can be achieved is 100

Good Prospects



The Utopian Space Agency are seeking to extend the regime's reach into the inner solar system. In order establish a viable colony on the Martian surface the agency is sending a set of Prospector Probes which will analyse the water and mineral concentrations in the soil up to the limits of their sensor strength.

For the purpose of the analysis the Martian surface has been projected onto a two-dimensional surface grid. Each surface grid cell element contains the following information for an area of the Martian surface.

Field	Details
Soil Water Percentage	Integer in the range [0 – 100]
Mineral Wealth	Real value in the range between some positive minimum and maximum value
Scanned	Boolean

Each probe is slightly different and is defined by the following:

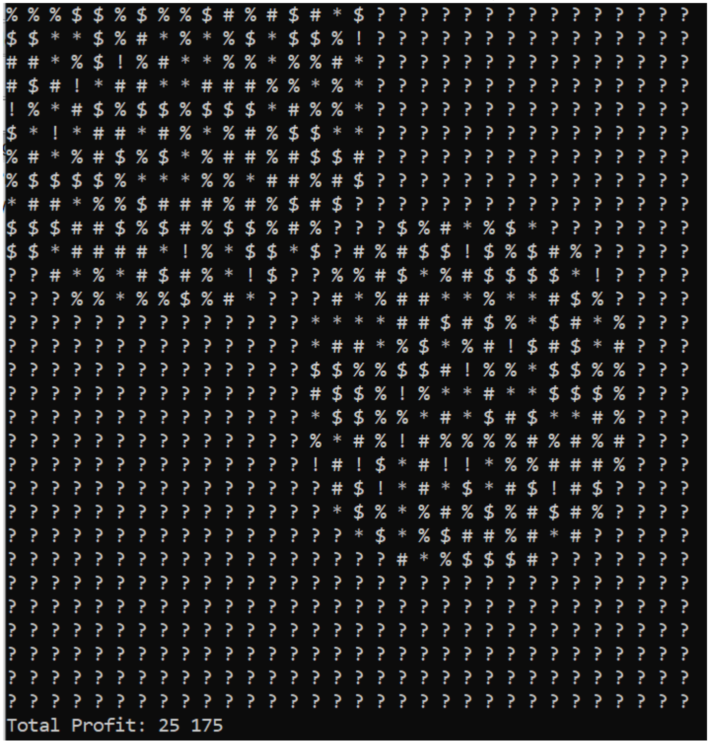
Field	Details
Sensor strength	Real value
Cost	Positive integer
Code	Sequence of alphanumeric characters of length 16
Coordinate	Structure contains a row and column value as a positive integer which indicates the probe's position on the surface

- You must create a simulation which will generate a two-dimensional data structure which stores generated areas of the Martian surface. All areas start with their scanned status as false and their other values must be in the ranges provided.
- The data structure must have a separate one-dimensional storage area for an arbitrary number of probes. Probes are placed here during the creation of the

Note: Although there are 120 marks available the maximum mark which can be achieved is 100

data structure. Probes are not placed in the 2D array directly but count as being “on the surface” for all calculations below.

- The Martian surface can be scanned by the probes. When that occurs the scanned status of all of the surface area grid cells which falls within the range of the probe are set to true. Different probes may have differently defined scanned behaviours. More than one probe may occupy the same coordinates when they are placed in the creation of the datastructure.
- Probes are assigned a location on creation and do not move thereafter.
- The results of the probe scan must be convertible into a text representation as follows.
    - Unscanned cells are represented as ‘?’ symbols
    - Scanned cells with a soil water percentage less than 5% are represented as ‘!’ symbols
    - Scanned cells more water than that minimum threshold are represented as follows:
        - ‘#’ symbols if the mineral wealth is in the range of 0 – 25% of the maximum
        - ‘%’ symbols if the mineral wealth is in the range of 25 – 50% of the maximum
        - ‘*’ symbols if the mineral wealth is in the range of 50 – 75% of the maximum
        - ‘$’ symbols if the mineral wealth is greater than 75%
    - The probes themselves contaminate the water table where they land so any cell containing a probe is displayed with the ‘@’ symbol.
- The wealth of all scanned cells with more than 5% soil water must be calculated and subtracted from the cost of the probes to give a total profit.
- An example of such output is given below where two probes both have a circular sensor radius (see T3 for more details):



```
% % % $ $ % $ % % $ # % # $ # * $ ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
$ $ * * $ % # * % * % $ * $ $ % ! ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
# # * % $ ! % # * * % % * % % # * ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
# $ # ! * # # * * # # # % % * % * ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
! % * # $ % $ $ % $ $ $ * # % % * ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
$ * ! * # # * # % * % # % $ $ * * ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
% # * % # $ % $ * % # # % # $ $ # ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
% $ $ $ $ % * * * % % * # # % # $ ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
* # # * % % $ # # # % # % $ # $ ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
$ $ $ # # $ % $ # % $ $ % # % ? ? ? $ % # * % $ * ? ? ? ? ? ? ?
$ $ * # # # # * ! % * $ $ * $ ? # % # $ $ ! $ % $ # % ? ? ? ? ?
? ? # * % * # $ # % * ! $ ? ? % % # $ * % # $ $ $ $ * ! ? ? ? ?
? ? ? % % * % % $ % # * ? ? ? # * % # # * * % * * # $ % ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? * * * * # # $ # $ % * $ # * % ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? * # # * % $ * % # ! $ # $ * # ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? $ $ % % $ $ # ! % % * $ $ % % ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? # $ $ % ! % * * # * * $ $ $ % ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? * $ $ % % * # * $ # $ * * # % ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? % * # % ! # % % % % # % # % # ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ! # ! $ * # ! ! * % % # # # % ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? # $ ! * # * $ * # $ ! # $ ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? * $ % * % # % $ % # $ # % ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? * $ * % $ # # % # * # ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? # * % $ $ $ # ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
Total Profit: 25 175
```

Note: Although there are 120 marks available the maximum mark which can be achieved is 100

You have been provided with the following in order to assist you in developing the system to the UJ standards:

- A `config.txt` file contains the configuration information for the simulation in the following format where EOL represents the end of line marker:
  - `ROWS COLS EOL`
  - `RANDOMSEED EOL`
  - `MAX_WEALTH MIN_WEALTH`
    A set of an arbitrary number of lines representing the details of the probes as follows:
  - `TYPE SENSOR_STRENGTH COST ROW COL EOL`
- `launchControl.dll` which contains the following C function with name mangling disabled: `void launch(int intCountDown);`

In order to meet the requirements of this task the project manager has broken the overall problem into several sub tasks (see accompanying mark sheet for more details):

- T0 – Design: You must create a UML 2.0 Class Diagram according to the requirements of the other tasks. It must be saved in a PDF file.
- T1 – Class Basics: You must create a class to represent the `MartianSurface`. The class must make use of the principles of good design known to you. It must additionally make use of a dynamic two-dimensional array of structures in which to store the surface data.
- T2 – The `MartianSurface` class **has a set of** `Probe`s which implement the `scan` method that will allow the areas of the `MartianSurface` to be revealed. The `MartianSurface` delegates its own `scan` method to the contained probes. The `MartianSurface` class must manage the life cycle of the `Probe` class.
- T3 – Modify the previously created classes to use an inheritance hierarchy with polymorphism as follows:
  - A `Probe` is an abstract base class with pure virtual functions called `scan` and `clone`
  - The `clone` function is overridden by the following derived classes to return a freestore allocated copy of the derived class as a `Probe*`
  - A `CircularProbe` is a kind of `Probe` that overrides the `scan` method to reveal the areas of the surface which fall within its radius defined by it's sensor strength
  - A `StochasticProbe` is a kind of `Probe` that overrides the `scan` method to potential reveal the areas of the entire surface randomly. The sensor strength value determines the likelihood that any area on the surface will be revealed by the scan.
  - The `MartianSurface's scan` method now must polymorphically call each of the contained `Probe's scan` function.
- T4 – Add the following to your `MartianSurface` class
  - An overloaded assignment operator which performs a deep copy of the underlying array.
  - Support for stream-based output of the `MartianSurface` as per the example provided
  - Support for double indexing by way of an overloaded function invocation operator.

Note: Although there are 120 marks available the maximum mark which can be achieved is 100

- T5
  - Define a `SurfaceCell` structure as per the given table with alignment padding disabled.
  - Add a member function to the `MartianSurface` class which will load the configuration information from the provided text file when the program starts.
  - Add a member function to `MartianSurface` which will save the contents of the two dimensional array to a binary file of records.
  - In all cases the functions will take the name of the file as a parameter and close the file when it has finished.
- Additionally you must create a `main.cpp` file which demonstrates the functionality of your class.
- Bonus
  - See descriptions on following page
  - Only one bonus may be attempted
  - The bonus must be submitted as a separate submission on EVE

Note: Although there are 120 marks available the maximum mark which can be achieved is 100

T0 - UML Class Diagram [10]

- Identified classes (2)
- Attributes (2)
- Operations (2)
- Has-a relationships (2)
- Is-a relationships (2)

T1 - Class Basics [20]

- Constructors (5)
- Destructors (5)
- Information hiding (5)
- Use of .h and .cpp files (3)
- Comments (2)

T2 - Composition & Delegation [20]

- Create contained class
- Has-a relationship (5)
- Lifecycle management (5)
- Delegation (5)
- Algorithm (5)

T3 - Inheritance & Polymorphism [20]

- Abstract base class (5)
- Overriding (5)
- Polymorphism (5)
- Member visibility (5)

T4 - Operator Overloading [15]

- Assignment = (5)
- Output << (5)
- Double Indexing ()(5)

T5 - File Handling [15]

- Define structure (5)
- Write binary file (5)
- Read text file (5)

You may only select <u>one</u> of the following bonus sections:

Bonus - Templates [20]

- Make the entire MartianSurface class templated and every member function must be implemented as a templated function.

Bonus - DLLs [20]

- Provide code for an executable which loads and runs the file from the given DLL using explicit linking (10) and implicit linking (5).
- Batch file (5) - The executables must both be compiled from a .BAT file

Bonus - Exception Handling [20]

- Use exception handling to handle errors involving numerical ranges using an inheritance hierarchy of exception classes.

Note: Although there are 120 marks available the maximum mark which can be achieved is 100