MI.103 Programming Techniques
Project – *Part 2*

The goal of this second part is to set up a generic maze with template classes.

Exercise 1: The template class GenericPoint2D

A generic point is represented by 2 coordinates (x and y). These coordinates can be integer, float, double,... Design and code the template class *GenericPoint2D*

Exercise 2: The template class GenericCell

A generic cell is represented by a genericPoint2D (e.g the center of the cell) and a flag. The flag is a boolean. Implement the template class *GenericCell*

Exercise 3: The template class Maze

A generic maze is a template abstract class and contains a matrix of GenericCell, an exit (which is represented by a *GenericPoint2D*) and a list of *GenericRobots*(see exercise 5)

Exercise 4: The concrete class GenericRandomMaze

Like the class *RandomMaze*, a generic random maze is a generic maze with random walls. This class inherite of the abstract class GenericMaze and contain two attributes :

— the number of the walls ;
— their maximal length.

Implement the concrete class GenericRandomMaze.

Exercise 5: The generic abstract class Robot

Like the class *Robot*, a generic robot is defined by a start position (e.g a *GenericPoint2D* and a list of covered position (e.g a list of *GenericPoint2D*. Like a robot, a generic robot has a method *go()*. This method depends on the robot's strategy to move.

Exercise 6: Generic crazy robot

Like the class *CrazyRobot*, this class models a generic crazy robot and inherits to the GenericRobot class. Implement the class *GenericCrazyRobot*.

Exercise 7: Maze test

Implement a main program to create a random maze and a crazy robot by considering the coordinates as float datatype.