|  |  |  |
| --- | --- | --- |
| **American University of Sharjah**  **College of Engineering**  Dept of Computer Science & Engg  P. O. Box 26666  Sharjah, UAE | A picture containing logo  Description automatically generated | **Instructors:** Dr. Aliaa Moualla  **Lab Instructor:** Sameer Alawnah  **Office:** EB1-0012C  **Phone**: 971-6-515-4940  **e-mail**: salawnah@aus.edu  **Semester**: Spring 2024 |

**CMP 220L - Programming II**

**Lab #12 – Heterogenous Collection and Exception Handling**

**Note: using ChatGPT will be considered a violation of the AUS integrity code.**

**Objectives:**

* Practice heterogeneous collection.
* Practice exception handling.

Using Visual Studio 2022, write the below programs, compile and provide screenshots of output.

Note: you are required to submit copy of the code + screenshots of program run for each exercise.

**Exercise #1**

A path element is an element you might find in a path. This could be a mountain, meadow, lake, or a path that is comprised of several elements. Write a C++ program that helps travelers plan their journeys by modeling the path.

* Create a class for **PathElement** which has a *distance*, and the following methods:
  + *Constructor*: to initialize the distance attribute.
  + *GetDistance*: returns the distance.
  + *Describe*: pure virtual void function.
  + *Draw*: pure virtual void function.
* Create classes for **Mountain**, **Meadow**, and **Lake** which are PathElements. For each class:
  + Implement *Describe* to print the name and the length (distance) of the element.
  + Implement *Draw* to print a visual representation of the element.
    - A mountain is a series of “/\”
    - A meadow is a series of “\_”
    - A lake is a series of “~”
* Write a main method that perform the following:
  + Declare a vector listPath of type PathElement pointers.
  + Add three instances of Mountain, Meadow, and Lake to the listPath vector using ‘new’ keyword.
  + Call the describe method for each element in the vector to print the information about each element.
  + Call the draw method for each element to visually draw them.
  + Remember to deallocate the memory.

A screenshot of a computer screen

Description automatically generated**Sample output**

#include<iostream>

#include<vector>

**using** **namespace** std;

**class** PathElement {

**public**:

PathElement()

{

distance = 0;

}

PathElement(**double** \_distance)

{

distance = \_distance;

}

**double** GetDistance()

{

**return** distance;

}

**virtual** **void** Describe() = 0;

**virtual** **void** Draw() = 0;

**protected**:

**int** distance;

};

**class** Mountain : **public** PathElement{

**public**:

Mountain(**int** \_distance) :PathElement(\_distance)

{

}

**void** Describe()

{

cout << "A mountain which is " << distance << "km long" << endl;;

}

**void** Draw()

{

**for** (**int** i = 0; i < distance; i++)

{

cout<<"/\\";

}

}

};

**class** Meadow : **public** PathElement {

**public**:

Meadow(**int** \_distance) :PathElement(\_distance)

{

}

**void** Describe()

{

cout << "A meadow which is " << distance << "km long"<<endl;

}

**void** Draw()

{

**for** (**int** i = 0; i < distance; i++)

{

cout << "\_";

}

}

};

**class** Lake : **public** PathElement {

**public**:

Lake(**int** \_distance) :PathElement(\_distance)

{

}

**void** Describe()

{

cout << "A lake which is " << distance << "km long"<<endl;

}

**void** Draw()

{

**for** (**int** i = 0; i < distance; i++)

{

cout << "~";

}

cout << endl;

}

};

**int** main()

{

vector<PathElement\*> listPath;

listPath.push\_back(**new** Mountain(4));

listPath.push\_back(**new** Meadow(5));

listPath.push\_back(**new** Lake(3));

**for** (**int** i = 0; i < listPath.size(); i++)

{

listPath[i]->Describe();

}

**for** (**int** i = 0; i < listPath.size(); i++)

{

listPath[i]->Draw();

}

**for** (**int** i = 0; i < listPath.size(); i++)

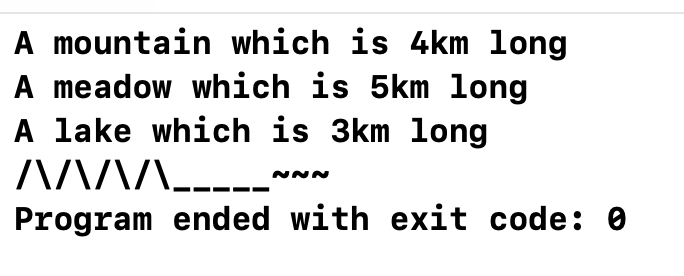
{

**delete** listPath[i];

}

**return** 0;

}

****