|  |  |  |
| --- | --- | --- |
| **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 #13 –Exception Handling**

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

**Objectives:**

* Practice exception handling.

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

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

A player has a name, a tool in their hand. A player spawns in the Overworld but can travel to the Nether and back. A player can sleep only in the Overworld.

Write a C++ program that models the Player. Define the following:

* A **Tool** which has the following values: IronSward, IronPickaxe, DiamondPickaxe
* A **Location** that has the values: Overworld, Nether
* A **State** that has the values: Asleep, Awake
* A **Player** class which has (all members are strings):
  + **Name**
  + **Location** (starting location is Overworld)
  + **State** (starting state is Awake)
  + **Hand** which stores the Tool that is currently in the player’s hand
  + ***Print*** function which prints the player’s information
  + ***Travel*** function which toggles the player’s location
  + ***Sleep*** function which changes the player’s state to asleep if the player isn’t already asleep and they’re not in the Nether. The function should throw an InvalidActionException otherwise. The program exits once the player goes to sleep successfully.
* **InvalidActionException** class which takes the action as an attribute

**Sample main**

int main() {

Player steve("Steve", "IronSward");

steve.print();

cout << "\n";

steve.travel();

cout << "------------Trying to sleep------------" << endl;

try {

steve.sleep();

}

catch (InvalidActionException e) {

cout << "Exception: " << e.what() << endl;

}

steve.travel();

cout << "------------Trying to sleep------------" << endl;

try {

steve.sleep();

}

catch (InvalidActionException e) {

cout << "Exception: " << e.what() << endl;

}

return 0;

}

**Sample output**

Name: Steve

Location: Overworld

State: Awake

Hand: IronSward

Traveled to the Nether

------------Trying to sleep------------

Exception: Invalid action - sleeping in the Nether

Traveled to the Overworld

------------Trying to sleep------------

Player is sleeping now

#include <iostream>

#include <string>

using namespace std;

class InvalidActionException: public exception {

private:

string action;

public:

InvalidActionException(string \_action)

{

action = \_action;

}

virtual const char\* what() const

{

return action.c\_str();

}

};

class Player {

public:

Player(string n, string h) {

name = n;

hand = h;

location = "Overworld";

state = "Awake";

}

void print() {

cout << "Name: " << name << endl;

cout << "Location: " << location << endl;

cout << "State: " << state << endl;

cout << "Hand: " << hand << endl;

}

void travel() {

if (location == "Overworld")

{

cout << "Traveling to nether";

location = "Nether";

}

else if (location == "Nether")

{

cout << "Traveling to overworld";

location = "Overworld";

}

}

void sleep() {

if (state == "asleep") {

cout << "Player is already asleep\n";

}

else

if (location == "Nether")

throw InvalidActionException("sleeping in the nether");

else {

state = "Asleep";

cout << "Player is sleeping now\n";

}

}

private:

string name, location, state, hand;

};

int main() {

Player steve("Steve", "IronSward");

steve.print();

cout << "\n";

steve.travel();

cout << "------------Trying to sleep------------" << endl;

try {

steve.sleep();

}

catch (InvalidActionException e) {

cout << "Exception: " << e.what() << endl;

}

steve.travel();

cout << "------------Trying to sleep------------" << endl;

try {

steve.sleep();

}

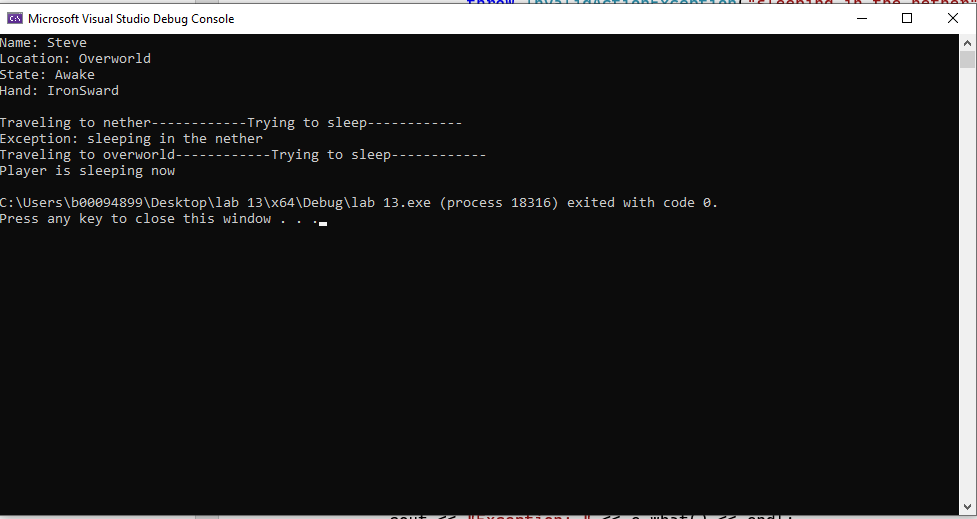
catch (InvalidActionException e) {

cout << "Exception: " << e.what() << endl;

}

return 0;

}

****