Problem 2B

A. Add a function named isEqual() to your Point3D class. The following code should run correctly:

Solution:

```
#include <iostream>

using namespace std;

class Point3D
{
public:
  void setValues(int _m_x, int _m_y, int _m_z)
  {
    m_x = _m_x;
    m_y = _m_y;
    m_z = _m_z;
  }

  void print()
  {
    cout << "<" << m_x << ", " << m_y << ", " << m_z << ">\n";
  }

  bool isEqual(Point3D classInstance)
  {
    return m_x == classInstance.m_x && m_y == classInstance.m_y && m_z ==
classInstance.m_z;
  }

private:
  int m_x;
  int m_y;
  int m_z;
};

int main()
{
  Point3D point1;
  point1.setValues(1, 2, 3);

  Point3D point2;
  point2.setValues(1, 2, 3);

  if (point1.isEqual(point2))
  {
    cout << "point1 and point2 are equal\n";
  }
```

```
  else
  {
    cout << "point1 and point2 are not equal\n";
  }

  Point3D point3;
  point3.setValues(3, 4, 5);

  if (point1.isEqual(point3))
  {
    cout << "point1 and point3 are equal\n";
  }
  else
  {
    cout << "point1 and point3 are not equal\n";
  }

  system("pause");
  return 0;
}
```