Fundamentals of Computer Programming
Lab 7 – OOP Intro

Spring 2019

1- Write a class named Point3d which contain:
   - Three private int members m_x, m_y, m_z
   - A constructor that sets the three points
   - A function called setValues() to set their values
   - A function that prints the point as <m_x, m_y, m_z>
   - A function isEqual() that takes another Point3d and check if they are equal
   - A function distanceTo() that takes another Point3d and returns the Euclidean distance between them

   .

2- Write a Rectangle class which contains:
   - Two private double members m_len, m_width
   - An enumeration Colors with the following values (‘BLACK’, ‘BLUE’, ‘RED’, ‘GREEN’, ‘WHITE’, MAX_COLOR_SIZE)
   - A const Color member m_color
   - A no argument constructor that defaults m_len = 1 and m_width = 1 and m_color = BLUE
   - A constructor that takes the length and defaults width and color
   - A constructor that sets all the members
   - A function getArea() that returns the area
   - A function getPerimeter() that returns the perimeter
   - A function getColor that prints the color of the rectangle (hint: use switch in a getColorName(Colors c) function)

3- Write a class named Array that dynamically allocate an int array with a length passed to the constructor. It should have the following functionality:
   - A function setValue(int index, int value) which does index checking. i.e. if I try to access and element out of the array it should return 0 as an error flag. Also make sure that value is positive otherwise return 0
   - A function getValue(int index) which also does index checking and returns -1 if the index is out of bounds
   - A function pushback(int value) that accepts only positive values and does index checking. If any error occur return 0 otherwise 1
   - A function pop() that return the value at the top of the stack counter and -1 if the stack is empty
   - Hint: you will need to save the length as well as use a stack counter member

4- Monster Generator:

- Create a Monster class that contain:
  - Enumerator MonsterType with the following types: ZOMBIE, VAMPIRE, SKELETON, DRAGON, MAX_MONSTER_TYPE
  - 3 members m_type, m_name, m_health
  - A constructor to initialize the three members
  - Print function that prints: “<m_name> is a <m_type> and has <m_health> left”
- A random int generator

```
int getRandomNumber(int min, int max)
{
static const double fraction = 1.0 / ((double) RAND_MAX + 1.0);
return (int)(rand() * fraction * (max - min + 1) + min);
}
```

- A getMonster() function that generates a random monster with:
  - A random type
    (use getRandomNumber(0, (int) Monster:: MAX_MONSTER_TYPE))
  - Random health between 1 and 100
  - Random name from a string array with 5 names of your choice
    for simplicity use the string array as a global variable