

CMPS 2143 Object Oriented Programming

Program 3: 100 points

DUE: Monday, Oct 23, 2017

Purpose: To declare/define/implement and use abstract classes, a base case and two *subclasses* demonstrating understanding of abstraction, inheritance typing and polymorphism.

Problem: *Spacecraft 2.0*

NASA2 was so thrilled with the success of the asteroid-collecting ship, they want to send more ships that can do other things in space! In addition to the asteroid-collecting ship, they want to build and send a smaller probe that can go to asteroids, “take pictures”, determine the material making up the asteroid and signal this data back to Earth. The ship you declare/define/used in Program 2 will become a *subclass* called SpaceShip – you will be moving some methods out of it. You will also have a *subclass* called SpaceProbe. THESE two subclasses WILL INHERIT FROM a base class SpaceCraft, which in turn will inherit from an abstract class SpaceObject. (Asteroid will also inherit from SpaceObject). The SpaceShip and the SpaceProbe have certain responsibilities in common: they can move, they can compute a distance to an asteroid, they can find the nearest “appropriate” asteroid, they can “visit” asteroids, keep track of the distance they travel, as well as signal back to Earth their state, that is

- location (x, y coordinates)
- total distance travelled
- current number of asteroids collected or current number of asteroids visited.

The SpaceShip (in addition to the common responsibilities that all spacecraft have) can determine which “not too-large”, not-picked-up asteroid is closest, then go pick up that asteroid.

The SpaceProbe (in addition to the common responsibilities that all spacecraft have) can determine which not-visited asteroid is closest, then go visit and collect data about that asteroid. In addition, it keeps a copy of all visited asteroids that are composed of precious metals, and can signal/provide this list to Earth.

You will need to add to your Asteroid class: asteroids know if they are composed of precious metals.

Your main program is to be a *driver* that tests the subclasses. You will still need to read in asteroids, though, to test the new and modified classes. As in Program 2, asteroids’ information is loaded from a file.

Input: See Program 2’s modified data file

Output: Your test driver should have the SpaceShip pick up the three closest asteroids (using the same algorithm from program 2) and have the SpaceProbe go out and visit 10 asteroids and determine if they are composed of precious metals. (Make sure you turn the ships on first.) *All* output goes to the console.

Turn in:

1. A UML class relationship diagram, with the subclasses, their member data and methods indicated.
(10 points toward homework grades - ***DUE Friday, October 6, 2017***)
2. My input data file for program 3
3. Console output from two execution runs.
4. Abstract and subclasses declarations or interfaces (header/interface files)
5. Abstract and subclasses implementations (implementation files)
6. Main program file.