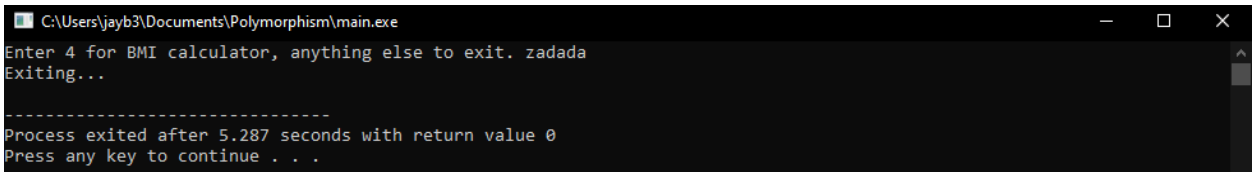
The two run-time binding methods added to the Polymorphism program are a BMI calculator and a “Happiness Test”, an advanced tool to measure the user’s happiness.

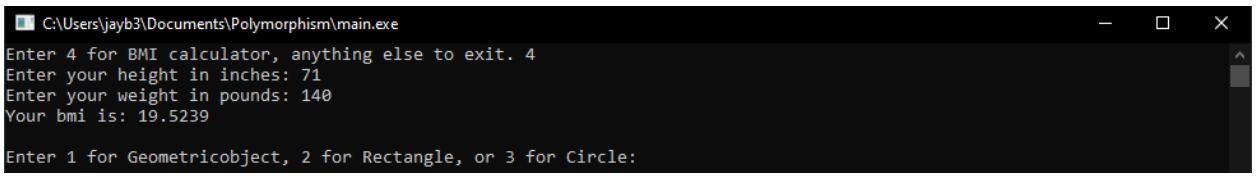
We will begin testing different inputs for the BMI calculator. To start the calculator, the user must input 4. Anything else will exit the program

Once we input 4, the program will begin.

If the user inputs anything else it will exit as stated

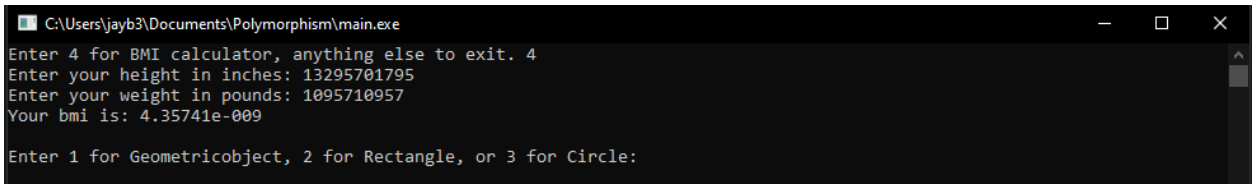


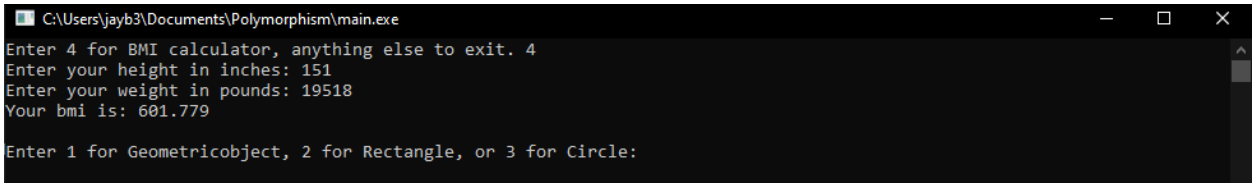
The program will then ask for a height in inches, and a weight in pounds. Each are a different input

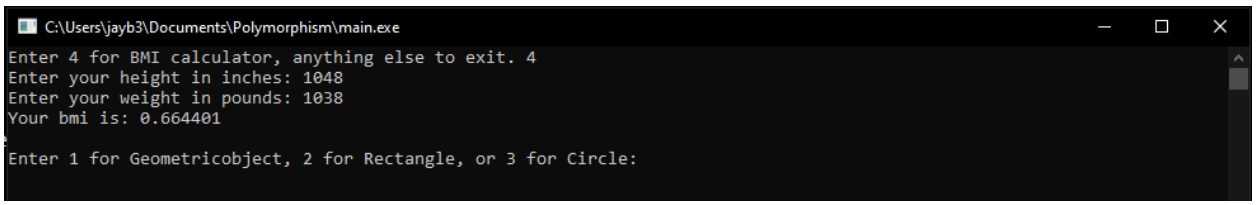


After both inputs are collected the BMI will be calculated and printed to the screen. The BMI uses the formula Weight / Height / Height * 703.

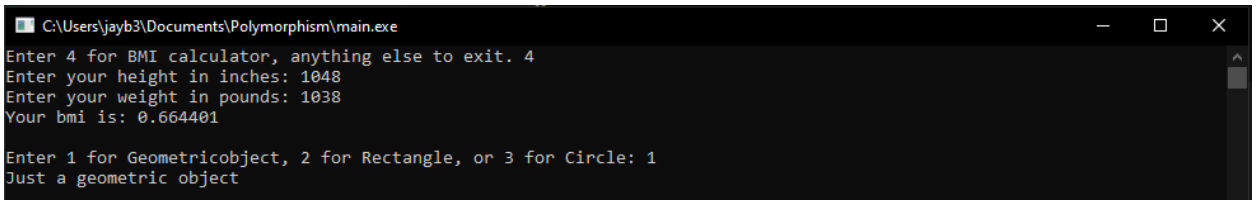
Various input tests:

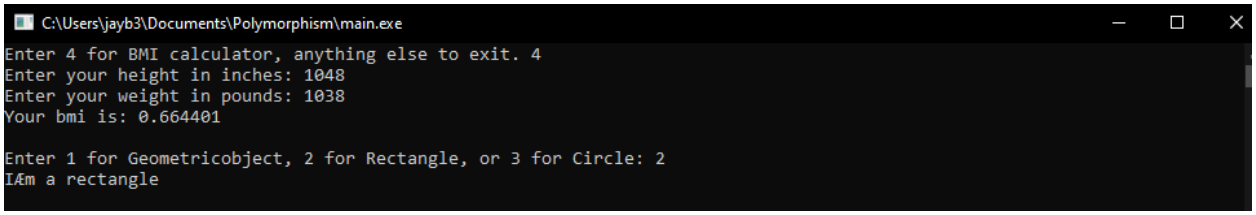


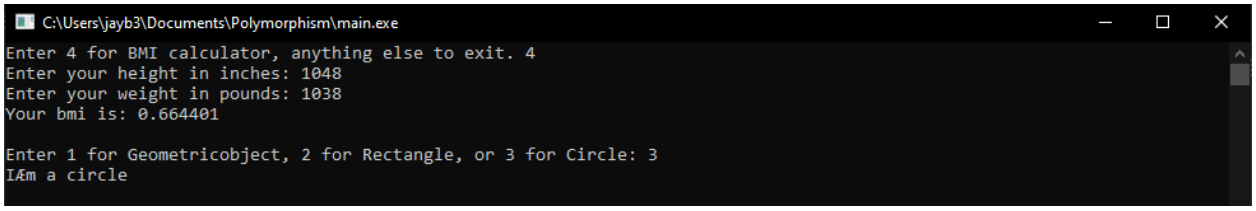


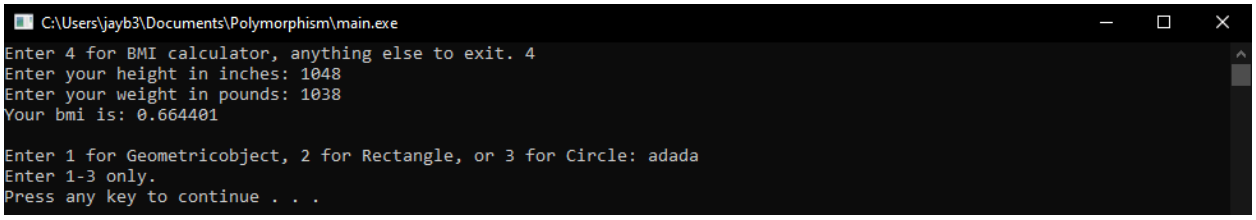


We also have our Geometric object, rectangle and circle. We learned about this in class already so I will keep this part brief.

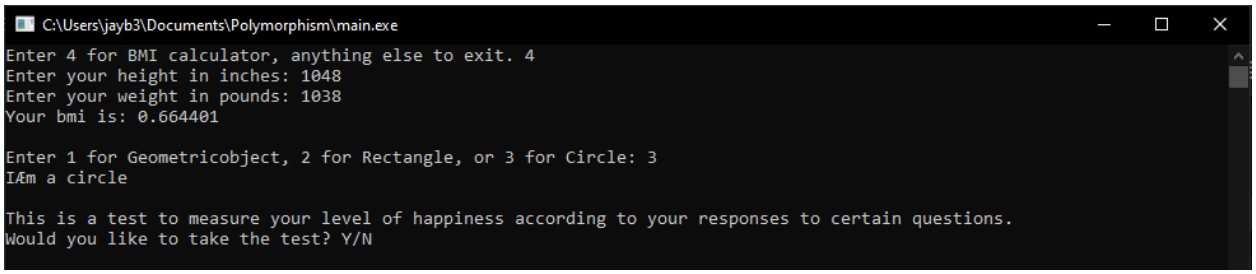


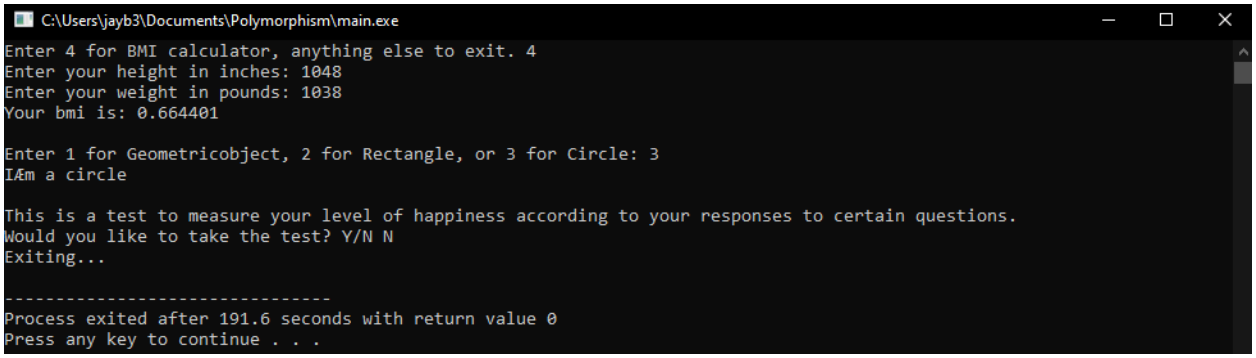






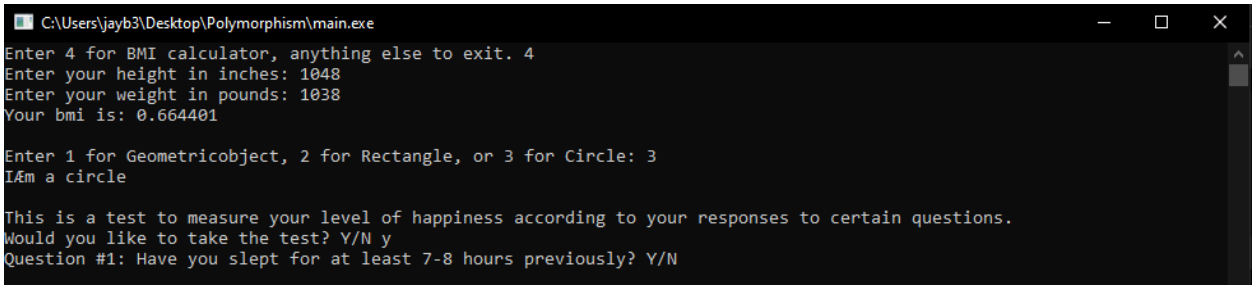
Next we have the “Happiness Test”. Like the Geometric run-time binding method, it starts right after the last method. The user must input Y to begin the test. N will exit the program.



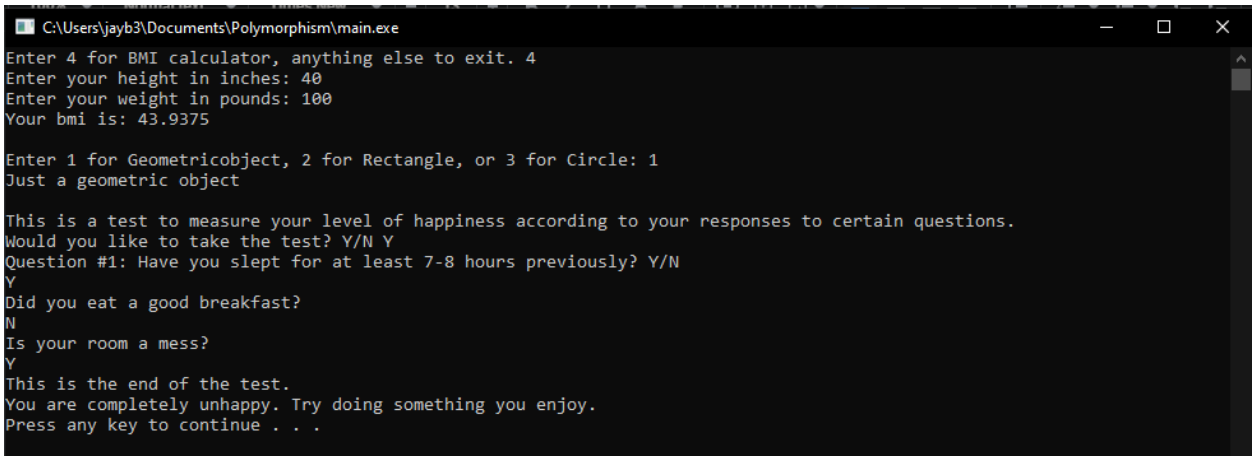


The test is simple for the user. They must enter Y/N in response to 3 questions. Specific responses to the questions give the user a “point” towards the integer variable “happynum” by incrementing it by 1.

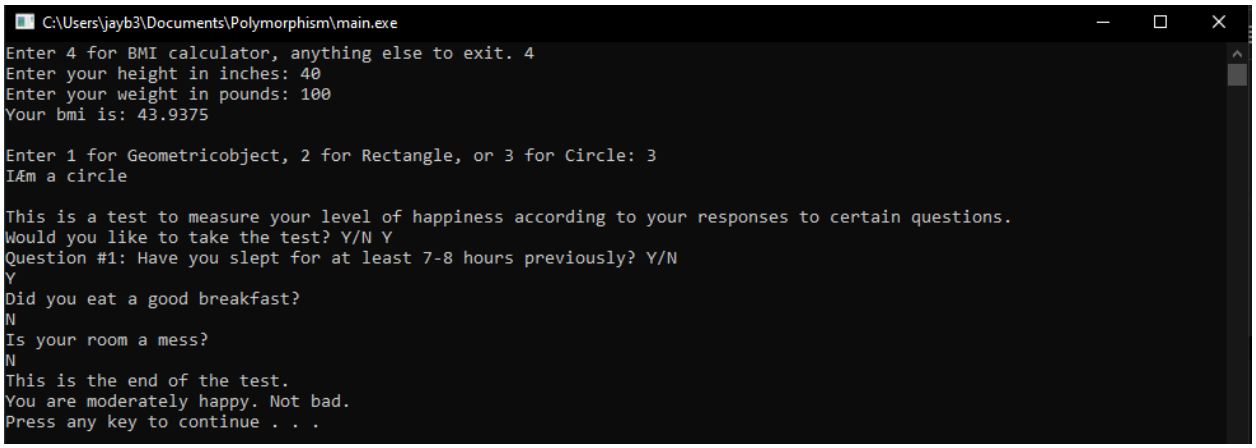
The program also accepts y/n as well for any Y/N question.



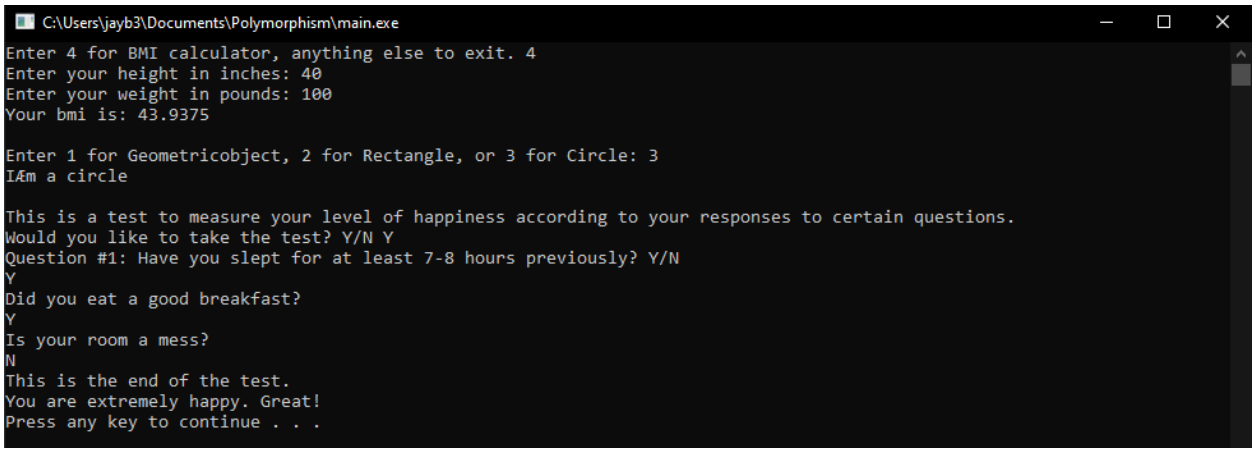
If the user chooses responses that denote happiness for ⅓ of the questions, the program will print the following:



If the user chooses responses that denote happiness for ⅔ of the questions, the program will print the following:



If the user chooses responses that denote happiness for 3⁄3 of the questions, the program will print the following:



That was a synopsis of the program. We achieved our goal of adding another 2 adding another two virtual run-time binding methods:

```
  ptr -> bmi(); //run-time binding
```

```
ptr -> happy(); //run-time binding
```

These methods both accept input from the user and make computations.