



# CMSC 150

# INTRODUCTION TO COMPUTING

LAB – WEEK 5

- MODULES

# PROGRAMMING WITH MODULES

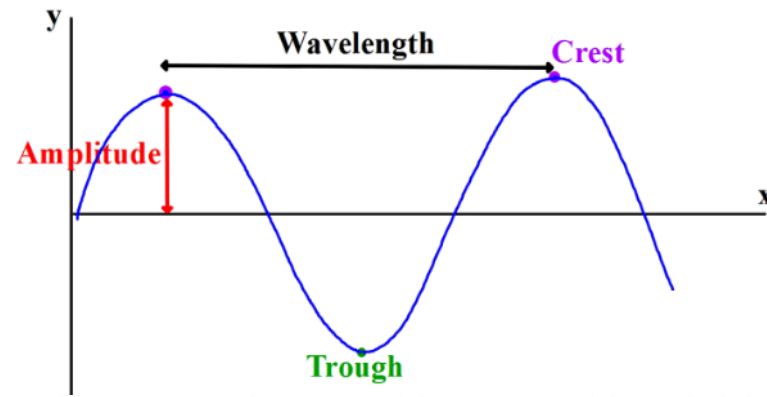
- Each module is contained within its own file
- Each module needs a unit test (its own main function)
  - Exercise each function you write in the module
- Break the problem down into manageable pieces!



The background features a light blue, concentric circular pattern. In the four corners, there are decorative circuit-like lines in a slightly darker blue, consisting of straight lines and small circles, resembling a stylized PCB or network diagram.

# EXERCISE – EVERYONE CODE ALONG

# EXERCISE



- Lets make a simple program to study the effects of a wave
- Equation

$$f(x) = A * \cos(Bx - C)$$

- $A$  is the amplitude/height of the wave
- $B$  determines the period of the wave (wavelength)
- $C$  determines the phase shift of the wave (where first crest occurs)

# EXERCISE

- Will make a plotting program to plot  $n$  points on the curve
- Assume  $x$  values range from -10 to 10
- Break the problem into manageable pieces!
  - Input –  $A, B, C, n$  (4 command line arguments)
  - Need to be able to compute an interval distance between each of the points
  - Need to be able to evaluate the function
  - Output – Plot of the curve
- Make two files
  - Wave.java – handles the math (library)
  - DrawWave.java – handles the graphics (client)



# EXERCISE WAVE.JAVA

- Wave needs to be able to calculate the interval between points, determine the position of the  $i$ th interval, and grab a specific value of the function
- Have a main function for unit testing

```
1. public class Wave {
2.
3.     /// @param n Number of points
4.     /// @return x interval between each point
5.     public static double GetInterval(int n) {
6.     }
7.
8.     /// @param i ith interval number
9.     /// @param interval x interval between each point
10.    /// @return ith interval x value
11.    public static double GetT(int i, double interval) {
12.    }
```

```
15.    /// @param amplitude A in  $A \cdot \cos(Bt - C)$ 
16.    /// @param period B in  $A \cdot \cos(Bt - C)$ , period is  $2\pi/B$ 
17.    /// @param phase C in  $A \cdot \cos(Bt - C)$ , phase shift is  $B/C$ 
18.    /// @param t time in parametric equation
19.    /// @return  $A \cdot \cos(Bt - C)$ 
20.    public static double GetValue(double amplitude,
21.    double period, double phase,
22.    double t) {
23.    }
24.    public static void main(String[] args) {
25.        //Simple test of wave module
26.    }
27. }
```

# EXERCISE WAVE.JAVA

- Wave needs to be able to calculate the interval between points, determine the position of the  $i$ th interval, and grab a specific value of the function
- Have a main function for unit testing

```
1. public class Wave {
2.
3.     /// @param n Number of points
4.     /// @return x interval between each point
5.     public static double GetInterval(int n) {
6.         return (10. - -10.) / n; ///[-10, 10] is hardcoded interval of values
7.     }
8.
9.     /// @param i ith interval number
10.    /// @param interval x interval between each point
11.    /// @return ith interval x value
12.    public static double GetT(int i, double interval) {
13.        return -10 + i * interval;
14.    }
```

```
15.    /// @param amplitude A in  $A*\cos(Bt - C)$ 
16.    /// @param period B in  $A*\cos(Bt - C)$ , period is  $2\pi/B$ 
17.    /// @param phase C in  $A*\cos(Bt - C)$ , phase shift is  $B/C$ 
18.    /// @param t time in parametric equation
19.    /// @return  $A*\cos(Bt - C)$ 
20.    public static double GetValue(double amplitude, double period, double phase,
21.        double t) {
22.        return amplitude * Math.cos(period*t - phase);
23.    }
24.
25.    public static void main(String[] args) {
26.        double amplitude = Double.parseDouble(args[0]);
27.        double period = Double.parseDouble(args[1]);
28.        double phase = Double.parseDouble(args[2]);
29.        int n = Integer.parseInt(args[3]);
30.
31.        //Simple test of wave module
32.        double interval = GetInterval(n);
33.        for(int i = 0; i < n; ++i)
34.            System.out.println(GetValue(amplitude, period, phase, GetT(i, interval)));
35.    }
36. }
```

# EXERCISE DRAWWAVE.JAVA

- DrawWave should be able to initialize, draw axis, draw wave

```
1. public class DrawWave {
```

```
2.
```

```
3. //Set up X scale, Y scale, and canvas size
```

```
4. public static void Initialize() {
```

```
5. }
```

```
6.
```

```
7. //Draw axis
```

```
8. public static void DrawAxis() {
```

```
9. }
```

```
18. /// @brief Draw wave
```

```
19. /// @param interval Interval between  
points
```

```
20. /// @param wave Wave values
```

```
21. public static void DrawWave(double  
amplitude, double period, double phase,  
int n) {
```

```
22. }
```

```
23.
```

```
24. public static void main(String args[]) {
```

```
25. //Read in wave parameters
```

```
26.
```

```
27. //Draw wave
```

```
28. }
```

```
29. }
```



# EXERCISE DRAWWAVE.JAVA

- DrawWave should be able to initialize, draw axis, draw wave

```
1. public class DrawWave {
2.
3.     //Set up X scale, Y scale, and canvas size
4.     public static void Initialize() {
5.         StdDraw.setCanvasSize(500, 500);
6.         StdDraw.setXscale(-11, 11);
7.         StdDraw.setYscale(-11, 11);
8.     }
9.
10.    //Draw axis
11.    public static void DrawAxis() {
12.        StdDraw.setPenColor(StdDraw.RED);
13.        StdDraw.line(-10, 0, 10, 0);
14.        StdDraw.textLeft(9.5, -0.5, "x");
15.        StdDraw.line(0, -10, 0, 10);
16.        StdDraw.textLeft(0.5, 9.5, "y");
17.    }
```

```
18.    /// @brief Draw wave
19.    /// @param interval Interval between points
20.    /// @param wave Wave values
21.    public static void DrawWave(double amplitude, double period, double
phase, int n) {
22.        StdDraw.setPenColor(StdDraw.BLACK);
23.        double interval = Wave.GetInterval(n);
24.        for(int i = 0; i < n; ++i) {
25.            double t = Wave.GetT(i, interval);
26.            StdDraw.point(t, Wave.GetValue(amplitude, period, phase, t));
27.        }
28.    }
29.
30.    public static void main(String args[]) {
31.        //Read in wave parameters
32.        double amplitude = Double.parseDouble(args[0]);
33.        double period = Double.parseDouble(args[1]);
34.        double phase = Double.parseDouble(args[2]);
35.        int n = Integer.parseInt(args[3]);
36.
37.        //Draw wave
38.        Initialize();
39.        DrawAxis();
40.        DrawWave(amplitude, period, phase, n);
41.    }
42. }
```

# EXERCISES

- Alter the API and implementation of your program to allow for an arbitrary range of data  $[min, max]$ 
  - Need to modify GetInterval, GetT, initialize, draw axis, draw wave
- Allow keyboard input to alter the initial wave parameters
  - Example: 'w' increases amplitude, 's' decreases amplitude, etc
- Continually animate the drawing of the wave!
- Augment function to also allow vertical shift by D
- Allow multiple waves

