1st Code Rendition:

```
package Mastery;

import java.util.Scanner;

public class ProjectMastery {

    public static void main(String[] args) {

        Scanner userInput = new Scanner(System.in);

        System.out.print("Designing: ");
        int design = userInput.nextInt();

        System.out.print("Coding: ");
        int code = userInput.nextInt();

        System.out.print("Debugging: ");
        int debug = userInput.nextInt();

        System.out.print("Testing: ");
        int test = userInput.nextInt();

        double designprcnt = ( (double) design / (design + code + debug + test)) * 100;
        double codeprcnt = ( (double) code / (design + code + debug + test)) * 100;
        double debugprcnt = ( (double) debug / (design + code + debug + test)) * 100;
        double testprcnt = ( (double) test / (design + code + debug + test)) * 100;

        System.out.println("Task % Time");
        System.out.println("Designing" + designprcnt);
        System.out.println("Coding" + codeprcnt);
        System.out.println("Debugging" + debugprcnt);
        System.out.print("Testing" + testprcnt);
    }

}
```

Output:

```
Designing: 100
Coding: 150
Debugging: 140
Testing: 110
Task % Time
Designing 20.0
Coding 30.0
Debugging 28.000000000000004
Testing 22.0
```

The first major issue I noticed was that my output was not formatted as the prompt wanted. To correct this I included proper spacing within the final 5 .print() commands.

2nd Code Rendition:

```java
package Mastery;

import java.util.Scanner;

public class ProjectMastery {

    public static void main(String[] args) {

        Scanner userInput = new Scanner(System.in);

        System.out.print("Designing: ");
        int design = userInput.nextInt();

        System.out.print("Coding: ");
        int code = userInput.nextInt();

        System.out.print("Debugging: ");
        int debug = userInput.nextInt();

        System.out.print("Testing: ");
        int test = userInput.nextInt();

        double designprcnt = ( (double) design / (design + code + debug + test)) * 100;
        double codeprcnt = ( (double) code / (design + code + debug + test)) * 100;
        double debugprcnt = ( (double) debug / (design + code + debug + test)) * 100;
        double testprcnt = ( (double) test / (design + code + debug + test)) * 100;

        System.out.println("Task        % Time");
        System.out.println("Designing   " + designprcnt);
        System.out.println("Coding      " + codeprcnt);
        System.out.println("Debugging   " + debugprcnt);
        System.out.print("Testing     " + testprcnt);
    }

}
```

Output:

```
Designing: 100
Coding: 150
Debugging: 140
Testing: 110
Task        % Time
Designing   20.0
Coding      30.0
Debugging   28.000000000000004
Testing     22.0
```

Here I noticed that my code could be simplified by creating a new variable titled totalTime, which would be equal to the value of all the times summed up.

3rd Code Rendition:

```
double designprcnt = ( (double) design / (design + code + debug + test)) * 100;
double codeprcnt = ( (double) code / (design + code + debug + test)) * 100;
double debugprcnt = ( (double) debug / (design + code + debug + test)) * 100;
double testprcnt = ( (double) test / (design + code + debug + test)) * 100;
```

To >

```
int totalTime = design + code + debug + test;

double designprcnt = ( (double) design / (totalTime)) * 100;
double codeprcnt = ( (double) code / (totalTime)) * 100;
double debugprcnt = ( (double) debug / (totalTime)) * 100;
double testprcnt = ( (double) test / (totalTime)) * 100;
```

Next I noticed the fact that my outputs were not rounded to 2 decimal places, and often had rounding errors. To fix this I implemented a new method which would display the % rounded to 2 decimal places.

4th Code Rendition:

```java
package Mastery;

import java.util.Scanner;
import java.text.DecimalFormat;

public class ProjectMastery {

    public static DecimalFormat round = new DecimalFormat("0.00");

    public static void main(String[] args) {

        Scanner userInput = new Scanner(System.in);

        System.out.print("Designing: ");
        int design = userInput.nextInt();

        System.out.print("Coding: ");
        int code = userInput.nextInt();

        System.out.print("Debugging: ");
        int debug = userInput.nextInt();

        System.out.print("Testing: ");
        int test = userInput.nextInt();

        int totalTime = design + code + debug + test;

        double designprcnt = ( (double) design / (totalTime)) * 100;
        double codeprcnt = ( (double) code / (totalTime)) * 100;
        double debugprcnt = ( (double) debug / (totalTime)) * 100;
        double testprcnt = ( (double) test / (totalTime)) * 100;

        System.out.println("Task        % Time");
        System.out.println("Designing   " + round.format(designprcnt));
        System.out.println("Coding      " + round.format(codeprcnt));
        System.out.println("Debugging   " + round.format(debugprcnt));
        System.out.print("Testing     " + round.format(testprcnt));
    }

}
```

Output:

```
Designing: 100
Coding: 150
Debugging: 140
Testing: 110
Task        % Time
Designing   20.00
Coding      30.00
Debugging   28.00
Testing     22.00
```

Finally, I made some minor changes to the output formatting, and included a display of the total time spent on the project.

5th Code Rendition:

```
package Mastery;

import java.util.Scanner;
import java.text.DecimalFormat;

public class ProjectMastery {

    public static DecimalFormat round = new DecimalFormat("0.00");

    public static void main(String[] args) {

        Scanner userInput = new Scanner(System.in);

        System.out.print("Designing: ");
        int design = userInput.nextInt();

        System.out.print("Coding: ");
        int code = userInput.nextInt();

        System.out.print("Debugging: ");
        int debug = userInput.nextInt();

        System.out.print("Testing: ");
        int test = userInput.nextInt();

        int totalTime = design + code + debug + test;

        double designprcnt = ( (double) design / (totalTime)) * 100;
        double codeprcnt = ( (double) code / (totalTime)) * 100;
        double debugprcnt = ( (double) debug / (totalTime)) * 100;
        double testprcnt = ( (double) test / (totalTime)) * 100;

        System.out.println("\nTotal time spent: " + totalTime + " minutes.");
        System.out.println("\nTask         % Time");
        System.out.println("Designing    " + round.format(designprcnt) + "%");
        System.out.println("Coding       " + round.format(codeprcnt) + "%");
        System.out.println("Debugging    " + round.format(debugprcnt) + "%");
        System.out.print("Testing      " + round.format(testprcnt) + "%");
    }

}
```

Output:

```
Designing: 100
Coding: 150
Debugging: 140
Testing: 110

Total time spent: 500 minutes.

Task        % Time
Designing   20.00%
Coding      30.00%
Debugging   28.00%
Testing     22.00%
```