**Student projects**

[General guidelines 3](#_Toc68028)

[Project 2-1: Student Registration 4](#_Toc68029)

[Project 2-2: Pay Check Calculator 5](#_Toc68030)

[Project 2-3: Tip Calculator 6](#_Toc68031)

[Project 2-4: Price Comparison 7](#_Toc68032)

[Project 2-5: Travel Time Calculator 8](#_Toc68033)

[Project 3-1: Letter Grade Converter 9](#_Toc68034)

[Project 3-2: Tip Calculator 10](#_Toc68035)

Project 3-3: Change Calculator ......................................................................................1 1

Project 3-4: Shipping Calculator .................................................................................... 12

Project 3-5: Table of Powers .......................................................................................... 13

Project 4-1: Even or Odd Checker .................................................................................1 4

Project 4-2: Hike Calculator ...........................................................................................1 5

Project 4-3: Feet and Meters Converter ......................................................................... 16

Project 4-4: Sales Tax Calculator .................................................................................. 17

Project 4-5: Dice Roller ..................................................................................................1 8

Project 4-6: Prime Number Checker ..............................................................................1 9

Project 5-1: Tax Calculator (Debug) .............................................................................. 20

Project 5-2: Guessing Game (Debug) ............................................................................2 1

Project 6-1: Prime Number Checker ..............................................................................2 1

Project 6-2: Wizard Inventory ......................................................................................... 23

Project 6-3: Contact Manager ........................................................................................ 24

Project 6-4: Quarterly Sales ...........................................................................................2 5

Project 6-5: Tic Tac Toe ................................................................................................. 26

Project 7-1: Pig Game Rules .........................................................................................2 7

Project 7-2: Wizard Inventory ......................................................................................... 28

Project 7-3: Contact Manager ........................................................................................ 29

Project 7-4: Monthly Sales .............................................................................................3 0

Project 7-5: Email List Cleaner ......................................................................................3 1

Project 8-1: Tip Calculator ............................................................................................. 32

Project 8-2: Wizard Inventory ......................................................................................... 33

Project 8-3: Contact Manager ........................................................................................ 34

Project 8-4: Monthly Sales .............................................................................................3 5

Project 9-1: Interest Calculator ......................................................................................3 6

Project 9-2: Monthly Payment Calculator .......................................................................3 7

Project 9-3: Sales Report ............................................................................................... 38

Project 9-4: Aircraft Fuel Calculator ...............................................................................3 9

Project 10-1: HTML Converter .......................................................................................4 0

Project 10-2: Email Creator ............................................................................................ 41

Project 10-3: Interest Calculator .................................................................................... 42

Project 10-4: Pig Latin Translator .................................................................................. 43

Project 11-1: Birthday Calculator ...................................................................................4 4

Project 11-2: Arrival Time Estimator .............................................................................. 45

Project 11-3: Future Value Logger ................................................................................. 46

Project 12-1: Game Stats ............................................................................................... 47

Project 12-2: Bird Counter .............................................................................................4 8

Project 12-3: Champion Counter ...................................................................................4 9

Project 12-4: Monthly Sales ...........................................................................................5 0

Project 13-1: Greatest Common Divisor ........................................................................5 1

Project 13-2: Tree Pattern .............................................................................................. 52

Project 13-3: Binary Search ...........................................................................................5 3

Project 13-4: Number Finder .......................................................................................... 54

Project 14-1: Rectangle Calculator ................................................................................5 4

Project 14-2: Card Dealer ..............................................................................................5 6

Project 14-3: Customer Viewer ......................................................................................5 7

Project 15-1: Rectangle or Square Calculator ................................................................ 58

Project 15-2: Roshambo ................................................................................................5 9

Project 15-3: Customer or Employee Creator ................................................................6 0

Project 15-4: Random Integer List ................................................................................. 61

Project 16-1: Blackjack ..................................................................................................6 2

Project 17-1: Customer Data Importer ........................................................................... 63

Project 17-2: Player Manager ........................................................................................6 4

Project 17-3: Task List ...................................................................................................6 5

Project 17-4: Product Manager ...................................................................................... 66

Project 18-1: Right Triangle Calculator .......................................................................... 67

Project 18-2: Preferences ..............................................................................................6 8

# Project 2-1: Student Registration

Create a program that allows a student to complete a registration form and displays a completion message that includes the user’s full name and a temporary password.

### Console

|  |
| --- |
| **Registration Form** |
| **First name: Eric**  **Last name: Idle**  **Birth year: 1934**    **Welcome Eric Idle!**  **Your registration is complete.** |
| **Your temporary password is: Eric\*1934** |

### Specifications

* The user’s full name consists of the user’s first name, a space, and the user’s last name.
* The temporary password consists of the user’s first name, an asterisk (\*), and the user’s birth year.
* Assume the user will enter valid data.

# Project 2-2: Pay Check Calculator

Create a program that calculates a user’s weekly gross and take-home pay.

### Console

|  |
| --- |
| **Pay Check Calculator** |
| **Hours Worked: 35**  **Hourly Pay Rate: 14.50**  **Gross Pay: 507.5 Tax Rate: 18%**  **Tax Amount: 91.35**  **Take Home Pay: 416.15** |

### Specifications

* The formula for calculating gross pay is: **gross pay = hours worked \* hourly rate**  The formula for calculating tax amount is: **tax amount = gross pay \* (tax rate / 100)**  The formula for calculating take home pay is: **take home pay = gross pay – tax amount**
* The tax rate should be 18%, but the program should store the tax rate in a variable so that you can easily change the tax rate later, just by changing the value that’s stored in the variable.
* The program should accept decimal entries like 35.5 and 14.25.
* Assume the user will enter valid data.
* The program should round the results to a maximum of two decimal places.

# Project 2-3: Tip Calculator

Create a program that calculates the tip and total for a meal at a restaurant.

### Console

|  |
| --- |
| **Tip Calculator** |
| **Cost of meal: 52.31**  **Tip percent: 20**    **Tip amount: 10.46** |
| **Total amount: 62.77** |

### Specifications

* The formula for calculating the tip amount is: **tip = cost of meal \* (tip percent / 100)**
* The program should accept decimal entries like 52.31 and 15.5.
* Assume the user will enter valid data.
* The program should round the results to a maximum of two decimal places.

# Project 2-4: Price Comparison

Create a program that compares the unit prices for two sizes of laundry detergent sold at a grocery store.

### Console

|  |
| --- |
| **Price Comparison** |
| **Price of 64 oz size: 5.99**  **Price of 32 oz size: 3.50**    **Price per oz (64 oz): 0.09** |
| **Price per oz (32 oz): 0.11** |

### Specifications

* The formula for calculating price per ounce is: **price per ounce = price / ounces**  Assume the user will enter valid data.
* The program should round the results to a maximum of two decimal places.

# Project 2-5: Travel Time Calculator

Create a program that calculates the estimated hours and minutes for a trip.

### Console

|  |
| --- |
| **Travel Time Calculator** |
| **Enter miles: 200**  **Enter miles per hour: 65**    **Estimated travel time**  **Hours: 3**  **Minutes: 5** |

### Specifications

* The program should only accept integer entries like 200 and 65.
* Assume that the user will enter valid data.

### Hint

 Use integers with the integer division and modulus operators to get hours and minutes.

# Project 3-1: Letter Grade Converter

Create a program that converts number grades to letter grades.

### Console

|  |
| --- |
| **Letter Grade Converter** |
| **Enter numerical grade: 90**  **Letter grade: A**    **Continue? (y/n): y**    **Enter numerical grade: 88**  **Letter grade: A**    **Continue? (y/n): y**    **Enter numerical grade: 80**  **Letter grade: B**    **Continue? (y/n): y**    **Enter numerical grade: 67**  **Letter grade: C**    **Continue? (y/n): y**    **Enter numerical grade: 59**  **Letter grade: F**    **Continue? (y/n): n**    **Bye!** |

### Specifications

* The grading criteria is as follows:
  1. **88-100**
  2. **80-87**
  3. **67-79**
  4. **60-66 F <60**
* Assume that the user will enter valid integers for the grades.
* The program should continue only if the user enters “y” or “Y” to continue.

# Project 3-2: Tip Calculator

Create a program that calculates three options for an appropriate tip to leave after a meal at a restaurant.

### Console

|  |
| --- |
| **Tip Calculator** |
| **Cost of meal: 52.31**    **15%**  **Tip amount: 7.85**  **Total amount: 60.16**    **20%**  **Tip amount: 10.46**  **Total amount: 62.77**    **25%**  **Tip amount: 13.08**  **Total amount: 65.39** |

### Specifications

* The program should calculate and display the cost of tipping at 15%, 20%, or 25%.
* Assume the user will enter valid data.
* The program should round results to a maximum of two decimal places.

## Project 3-3: Change Calculator

Create a program that calculates the coins needed to make change for the specified number of cents.

### Console

|  |
| --- |
| **Change Calculator** |
| **Enter number of cents (0-99): 99**    **Quarters: 3**  **Dimes: 2**  **Nickels: 0**  **Pennies: 4**    **Continue? (y/n): y**    **Enter number of cents (0-99): 55**    **Quarters: 2**  **Dimes: 0**  **Nickels: 1**  **Pennies: 0**    **Continue? (y/n): n**    **Bye!** |

### Specifications

* The program should display the minimum number of quarters, dimes, nickels, and pennies that one needs to make up the specified number of cents.
* Assume that the user will enter a valid integer for the number of cents.
* The program should continue only if the user enters “y” or “Y” to continue.

## Project 3-4: Shipping Calculator

Create a program that calculates the total cost of an order including shipping.

### Console

|  |
| --- |
| **===============================================================** |
| **Shipping Calculator**  **===============================================================**  **Cost of items ordered: 49.99**  **Shipping cost: 7.95**  **Total cost: 57.94**    **Continue? (y/n): y**  **===============================================================**  **Cost of items ordered: -65.50**  **You must enter a positive number. Please try again.**  **Cost of items ordered: 65.50**  **Shipping cost: 9.95**  **Total cost: 75.45**    **Continue? (y/n): n**  **===============================================================**  **Bye!** |

### Specifications

* Use the following table to calculate shipping cost:

**COST OF ITEMS SHIPPING COST**

**==============================**

**< 30.00 5.95**

**30.00-49.99 7.95**

**50.00-74.99 9.95**

**> 75.00 FREE**

* If the user enters a number that’s less than zero, display an error message and give the user a chance to enter the number again.

## Project 3-5: Table of Powers

Create a program that displays a table of squares and cubes for the specified range of numbers.

### Console

|  |  |
| --- | --- |
| **Table of Powers** |  |
| **Start number: 90**  **Stop number: 100**    **Number Squared** | **Cubed** |
| **====== =======**   1. **8100 729000** 2. **8281 753571** 3. **8464 778688** 4. **8649 804357** 5. **8836 830584** 6. **9025 857375** 7. **9216 884736** 8. **9409 912673** 9. **9604 941192** 10. **9801 970299** 11. **10000 1000000** | **=====** |

### Specifications

* The formulas for calculating squares and cubes are:

**square = x \*\* 2 cube = x \*\* 3**

* Use tabs to align the columns.
* Assume that the user will enter valid integers.
* Make sure the user enters a start integer that’s less than the stop integer. If the user enters a start integer that’s greater than the stop integer, display an error message and give the user a chance to enter the integers again.

## Project 4-1: Even or Odd Checker

Create a program that checks whether a number is even or odd.

### Console

|  |
| --- |
| **Even or Odd Checker** |
| **Enter an integer: 33 This is an odd number.** |

### Specifications

* Store the code that gets user input and displays output in the main function.
* Store the code that checks whether the number is even or odd in a separate function.
* Assume that the user will enter a valid integer.

## Project 4-2: Hike Calculator

Create a program that converts the number of miles that you walked on a hike to the number of feet that you walked.

### Console

|  |
| --- |
| **Hike Calculator** |
| **How many miles did you walk?: 4.5** |
| **You walked 6840 feet.** |

### Specifications

* The program should accept a float value for the number of miles.
* Store the code that gets user input and displays output in the main function.
* There are 1520 feet in a mile.
* Store the code that converts miles to feet in a separate function. This function should return an int value for the number of feet.
* Assume that the user will enter a valid number of miles.

## Project 4-3: Feet and Meters Converter

Create a program that converts feet to meters and vice versa.

### Console

|  |
| --- |
| **Feet and Meters Converter** |
| **Conversions Menu:**   1. **Feet to Meters** 2. **Meters to Feet**   **Select a conversion (a/b): a**    **Enter feet: 100**  **30.48 meters**    **Would you like to perform another conversion? (y/n): y**    **Conversions Menu:**   1. **Feet to Meters** 2. **Meters to Feet**   **Select a conversion (a/b): b**    **Enter meters: 100**  **328.08 feet**    **Would you like to perform another conversion? (y/n): n**    **Thanks, bye!** |

### Specifications

* The formula for converting feet to meters is: **feet = meters / 0.3048**
* The formula for converting meters to feet is: **meters = feet \* 0.3048**
* Store the code that performs the conversions in functions within a module. For example, store the code that converts feet to meters in a function in a module.
* Store the code that displays the title in its own function, and store the code that displays the menu in its own function, but store the rest of the code that gets input and displays output in a main function.
* Assume the user will enter valid data.
* The program should round results to a maximum of two decimal places.

## Project 4-4: Sales Tax Calculator

Create a program that uses a separate module to calculate sales tax and total after tax.

### Console

|  |
| --- |
| **Sales Tax Calculator** |
| **ENTER ITEMS (ENTER 0 TO END)**  **Cost of item: 35.99**  **Cost of item: 27.50**  **Cost of item: 19.59**  **Cost of item: 0**  **Total: 83.08**  **Sales tax: 4.98**  **Total after tax: 88.06**    **Again? (y/n): y**    **ENTER ITEMS (ENTER 0 TO END)**  **Cost of item: 152.50**  **Cost of item: 59.80**  **Cost of item: 0**  **Total: 212.3**  **Sales tax: 12.74**  **Total after tax: 225.04**    **Again? (y/n): n**    **Thanks, bye!** |

### Specifications

* The sales tax rate should be 6% of the total.
* Store the sales tax rate in a module. This module should also contain functions that calculate the sales tax and the total after tax. These functions should round the results to a maximum of two decimal places.
* Store the code that gets input and displays output in another module. Divide this code into functions wherever you think it would make that code easier to read and maintain.
* Assume the user will enter valid data.

## Project 4-5: Dice Roller

Create a program that uses a function to simulate the roll of a die.

### Console

|  |
| --- |
| **Dice Roller** |
| **Roll the dice? (y/n): y**    **Die 1: 3**  **Die 2: 6**  **Total: 9**    **Roll again? (y/n): y**    **Die 1: 1**  **Die 2: 1**  **Total: 2**  **Snake eyes!**    **Roll again? (y/n): y**    **Die 1: 6**  **Die 2: 6**  **Total: 12**  **Boxcars!**    **Roll again? (y/n): n** |

### Specifications

* The program should roll two six-sided dice.
* Store the code that rolls a single die in a function.
* Store the code that gets input and displays output in the main function. Whenever it’s helpful, use helper functions to split this code into other functions.
* The program should display a special message for two ones (snake eyes) and two sixes (boxcars).

## Project 4-6: Prime Number Checker

Create a program that checks whether a number is a prime number and displays the total number of factors if it is not a prime number.

### Console

|  |
| --- |
| **Prime Number Checker** |
| **Please enter an integer between 1 and 5000: 1 Invalid integer. Please try again.**  **Please enter an integer between 1 and 5000: 2**   1. **is a prime number.**     **Try again? (y/n): y**    **Please enter an integer between 1 and 5000: 3**   1. **is a prime number.**     **Try again? (y/n): y**    **Please enter an integer between 1 and 5000: 4**   1. **is NOT a prime number.**   **It has 3 factors.**    **Try again? (y/n): y**    **Please enter an integer between 1 and 5000: 6**  **6 is NOT a prime number.**  **It has 4 factors.**    **Try again? (y/n): n**    **Bye!** |

### Specifications

* A prime number is only divisible by two factors (1 and itself). For example, 7 is a prime number because it is only divisible by 1 and 7.
* If the number is not a prime number, the program should display its number of factors. For example, 6 has four factors (1, 2, 3, and 6).
* Store the code that gets a valid integer for this program in its own function.
* Store the code that calculates the number of factors for a number in its own function.
* Store the rest of the code that gets input and displays output in the main function.