Homework 2
Basics of C++ and Operators

CSI-140 Introduction to Programming

Instructor: Dr. Vikas Thammanna Gowda Semester: Fall 2025

Points: 100 **Assigned Date:** 09/16/2025 **Due Date:** 09/23/2025 (11:59 PM)

Name: _____ **Section:** _____

Failure to follow the instructions and submission guidelines may result in a reduction of up to 100% of the points.

Instructions

- **Handwritten:**
  - Write your name and course section.
  - Use the handout provided by the instructor to complete the HW. (There will be a few additional copies in the file folder outside my office door — West Hall 200.)
  - Print a blank template from Canvas and write on it by hand.
  - Use the soft copy from Canvas and write directly on a tablet. Submit a single PDF file.
- **Use of word processor:**
  - Add your name and course section.
  - The homework should be answered in chronological order.
  - Each question must be added in **bold** before answering.
  - Submission must be a single PDF file.
- **Why PDF?** PDF files are universally compatible, meaning they can be opened and viewed on virtually any device with a PDF reader. This makes them ideal for sharing documents with a wide range of recipients, regardless of their software or hardware.
- **Individual Work:** This is an individual homework assignment. While you are encouraged to discuss the problem and possible approaches with your classmates, all work must be completed independently.
- **Plagiarism Policy:** Any form of plagiarism — including copying code, solutions, or text from another student, use of AI to generate report/answers — will be considered academic dishonesty and will be reported according to college policy.
- **Late submission policy:** 50% penalty for late submissions within 1 week; no credit after 1 week unless prior arrangements made.

Submission Guidelines:

- Drop off your completed assignment in the file folder outside my office door (West Hall 200) or turn it in during lecture sessions.
- **All other submissions must be a single, PDF file that is clear and easy to read.**

Rubric

Criteria	Points	Grade
Name and Section are present	10	
Part 1: Datatypes are correctly identified (1.5 Points each)	12	
Part 2: Appropriate message is present for each variable (2 Points each)	16	
Part 3: Output is recorded with an appropriate message (2 Points each)	14	
Part 3: No logical errors (2 Points each)	14	
Part 4: Output is recorded with an appropriate message	14	
Part 4: No logical errors	20	

Build a Pet Simulator Using Variables and Operators

Follow these steps to deepen your understanding of variables, data types, and operators. You'll work with the provided `pet_stats_calculator.cpp` starter code and answer each question by editing, running, observing, and documenting the results.

Goal

In this homework, you will build a small pet simulator program in C++. You will practice:

- Declaring variables of different datatypes.
- Changing their values using arithmetic operators (`+, -, *, /, %`).
- Using compound assignment operators (`+=, -=, *=`).
- Using comparison operators (`>, <, >=, <=, !=`).
- Printing results to the screen with `cout`.

By the end, your program will start with a pet's stats, update them step by step, and finally show a summary of how your pet is doing.

Part 1: Initial State Analysis

Create the following variables with exact initial values:

- `pet_name = "Buddy"`
- `pet_type = 'D'`
- `age = 2, health = 85, energy = 40`
- `happiness = 7.5, wellness_score = 0.0`
- `is_sleeping = false`

Identify the datatypes of each variable: (1.5 Points each)

Identifier	Datatype
`pet_name`	
`pet_type`	
`age`	
`health`	
`energy`	
`happiness`	
`wellness_score`	
`is_sleeping`	

PART 2: Display initial values stored in each variable with an appropriate message and record the output below: (2 Points each)

Part 3: Update stats using arithmetic operators

PART 3a:

The feeding action should:

- Add 10 to `health`
- Add 5 to `energy`
- Add 1.5 to `happiness`

Display the values stored in each of the above variables with an appropriate message and record the output below: (4 Points each)

PART 3b:

- Subtract 10 from `energy`
- Add 2.0 to `happiness`

Display the values stored in each of the above variables with an appropriate message and record the output below: (4 Points each)

PART 3c:

The sleep action should:

- Multiply `energy` by 2
- Set `is_sleeping` to `true`

Display the values stored in each of the above variables with an appropriate message and record the output below: (4 Points each)

PART 4: Use Other Operators

PART 4a:

Create an integer called `age_in_days` and get user input for the pet's age in days. Calculate days until next birthday using modulo (%) operator. Assume there are 365 days in a year.

Display the pet's age in days and days until next birthday with an appropriate message and record the output below: (14 Points)

Part 4b:

Update the `wellness_score` according to the formula:

$$(\texttt{health} * 0.4) + (\texttt{happiness} * 3.0) + (\texttt{energy} * 0.6).$$

Display the new `wellness_score` with an appropriate message and record the output below: (10 Points)

Part 4c:

Calculate average of all three stats: `health`, `energy`, and `happiness`.
Display the average stats with an appropriate message and record the output below: (10 Points)