

Programming Principles Task 2

Programming Problem

Bart Blubber has invented a new rubber bouncing ball that when dropped from any height will bounce back to exactly three quarters ($\frac{3}{4}$) of that height and for every successive bounce it will bounce back to three quarters of the previous height until it eventually stops. This new ball also has a unique property, which is that if it is dropped from a height of one centimeter (1 cm) or less it will not bounce at all.

Write a program that will ask the user for a height from which the ball will be dropped. The entered height must be less than or equal to ten metres (10m or 1000cm). The program should then calculate **the total distance the ball will bounce** and **how many times the ball will bounce** when dropped from the **specified height**. Your program must **get the height** from the user.

The program should therefore be capable of calculating the;

- Number of times the ball will bounce when dropped from a specified height
- Total distance the ball will travel when it comes to rest after being dropped from the specified height.

To achieve a higher mark in the **producing aspect**, your program should also;

- ★ Enable the user to repeat the program until the user wishes to quit
- ★ Check that the entered height is less than or equal to 10m or 1000cm

For students who need a challenge you may also check that the entered height is numeric and doesn't contain alpha characters ie the input only contains digits 0 to 9.

Plan and Write This Program.

You must use the **Technology Process** to design and develop your program. See the table below on how you can address the Technology Process. You are required to include the following in your submitted project;

Investigating

- Define The Problem by producing an IPO diagram.

Devising

- Sketch a user interface form showing the visual basic objects you will use
- Show the properties you will change for each object
- Write a solution algorithm for each button, in pseudocode, which should include the three control structures – sequence, selection and iteration.

Producing

- Create the program in Visual Basic 6.
- A copy of the Source code written in Visual Basic (ie the .vbp and .frm files) and stored in a folder called "**Bouncing Ball**" in your home data.
- Program asks the user for an input height.
- Program tests for valid entered heights ($>0.01\text{m}$ or 1cm and $\leq 10\text{m}$ or 1000cm).
- Program calculates the total distance the ball bounces.
- Program calculates the number of times the ball bounces.
- Program enables the user to repeat the program until the user wishes to quit.

- Use good programming techniques such as;
 - ♦ Internal Documentation
 - ♦ Indenting when using selection and iteration control structures
 - ♦ Declared variables with meaningful names
- Create an executable program stored in the folder called “**Bouncing Ball**” in your home data.

Evaluating

Discussion in your evaluation;

- Difficulties you had and how they were overcome.
- Good and bad features of your program.
- If you were to produce program version 2 what changes or improvements would you make and why.
- How you tested your program for correctness of results.

Assessment

The table below is a guide to what is required in addressing the Technology Process.

Technology Process	Aspect Requirement	Structured Programming Design Tools Used
Investigating	<ul style="list-style-type: none"> ▪ Define The Problem 	IPO Diagram
Devising	<ul style="list-style-type: none"> • Sketch a user interface form showing the visual basic objects you will use. • Show the properties you will change for each object. ▪ Write a Solution Algorithm in pseudocode for each button 	<ul style="list-style-type: none"> ▪ Structure Theorem (Selection control Structures) ▪ Pseudocode Algorithm ▪ Stepwise refinement
Producing	<ul style="list-style-type: none"> • Code the Algorithm Into a Specific Programming Language ▪ Test the Program on the Computer. ▪ Program checks for errors. ○ Create an Executable 	<ul style="list-style-type: none"> • Control Structures ▪ Internal Documentation
Evaluating	<ul style="list-style-type: none"> • Discussion on how program was tested for correctness. ▪ Document the Program ○ Justify changes of final product compared to original design 	<ul style="list-style-type: none"> ○ External Documentation (User Manuals) ○ Help Systems

Due No Later Week 8