JustBASIC

KS3 Programming Workbook

*Follow the exercises in this booklet and log your progress on the back pages*

Name Form

# JustBASIC

## Starting out – Project 1

### The HelloWorld Program

Every programmer starts with the Hello World Program. The purpose of this is to write a program that will display a simple message on the screen.

We will need to use the **PRINT** Command

Command: PRINT – Displays a message on the screen

Usage: PRINT “Your Message”

To write this program, load up JustBASIC and start a new program (File 🡪 New)

Type this code in:

1 PRINT “Hello World”

2 End

To run (or *execute*) your program, click ***Run***

Save your program as HelloWorld

**Now Try These:**

1. Write a program to write your name on the screen

*Save as MyName*

1. Write a program to write the words to Twinkle Twinkle Little Star to the screen

Save as *Twinkle*

**Questions:**

What is the purpose of the PRINT Command?

What is the purpose of the END Command?

## Project 2 – Input

### The HowOldAreYou Program

We can make our programs more useful if we allow people to enter information into them. We can do this by using the **INPUT** command. This command prompts the user to enter a piece of information, which the program stores inside a **variable**

A variable is like a box that can be used to store a piece of information. We can put different information into the box. We will call our variable age because it will store a person’s age.

We will need to use the **INPUT** Command

Command: INPUT – Prompts the user to enter a piece of information

Usage: INPUT “A piece of text to display” ; VariableName

OR… INPUT VariableName

Start a new program and type in this code:

1 PRINT “Welcome to my program”

2 INPUT “Please enter your age: “ ; Age

3 PRINT Age; “ is a great age to be!”

4 End

*Save as HowOldAreYou*

Let’s take a closer look at this program:

**Line 1** is another simple use of the PRINT command

**Line 2** is our new INPUT command – it is made of two parts, one part places writing on the screen, and the other tells us that whatever number the user enters should be stored in a box (variable) called Age

**Line 3** is a new way of using the PRINT command. We are using it to put two things on the screen, the contents of the age variable and some more writing. These two items are separated by a semi colon. Notice how the variable does not need any quotation marks, but the writing does?

**Line 4** is another example of the End command – it tells the computer there are no more lines to run

**Now Try These:**

1. Write a program that asks for your favourite number, and then replies that it is its favourite number too

*Save as faveNum*

1. Write a program that asks for 2 numbers and then says that the second number is its preferred number

*Save as FaveNum2*

## Project 3 – Calculations

### The VATCalculator Program

In this project you will create a program to calculate the amount of VAT on a price. We will assume the VAT rate is 20%.

* The program will need to ask for the price to be entered
* It must then work out 20% of the number that has been entered
* It should then write the answer on the screen.

Start a new program and type this in:

1. PRINT “Welcome to the VAT Calculator Program”
2. INPUT “Type in the Price: “ ; Price
3. PRINT “The VAT on £”; Price ; “ is £” ; Price \* 0.2
4. END

*Save as VATCalculator*

Let’s take a closer look at the new ideas used in this program:

**Line 3** now combines 4 pieces of writing. It consists of a piece of text, and then the variable, another piece of text, and then a calculation (the variable multiplied by 0.2)

Notice that we used the asterisk symbol (\*) to multiply the price by the VAT

We can also use these mathematical symbols:

+ Add

- Multiply

/ Divide

**Now Try These:**

1. Adjust your VATCalculator Program to also display the total amount *including* the VAT

*Save as VATCalculator2*

1. Create a program to work out the area of a room. It should ask for the length, then the width, and then multiply them together to get the area

*Save as RoomArea*

1. Create a program to calculate the length of fencing required for a sheep pen. It should ask how wide and long the pen is to be, and then calculate the total length of fencing needed

*Save as SheepPen*

1. Write a program to calculate the amount of turf (grass) needed for a garden if a 1m border is left all the way around. It will need to ask for the length and width of the garden

*Save as TurfCalculator*

**Questions:**

Explain in your own words what is meant by the term **Variable**

## Project 4 – Working with Strings

### The HelloYou Program

In this program we will extend our earlier HelloWorld program to make it more personal. We will do this by prompting the user to enter their name.

To do this, we need to understand **data types**.

So far, we have created variables that contain numeric data (numbers) such as an age, or amount of money. We can also use a data type known as string to store text information.

Strings are nothing new. In the previous program, we have used string values every time we wanted to write text. We indicate to the computer that it is a string by enclosing it in quotation marks.

To create a string variable, we must add a dollar sign ($) to the variable name. When we create a variable to contain a string, we do not need to use quotation marks whenever we want to print that variable.

Start a new program and type in this code:

1. INPUT “What is your name? “ ; name$
2. PRINT “Hello “; name$; “. I am very pleased to meet you.”
3. END

*Save as HelloYou*

Notice that this time we were able to enter text instead of a number when the input command was used. The text we entered was then displayed on screen with the word Hello to make our program much more personal.

**Now Try these:**

1. Write a program that you can have a simple conversation with. It should ask various questions and have an appropriate response ready for when an answer is given

*Save as Conversation1*

1. Write a program that asks for the name of someone whose birthday is today, and then prints the lyrics to “Happy Birthday to You” with the person’s name entered in the appropriate place on line 3.

*Save as Birthday*

1. Write a program that asks for your name, and then your age. It should reply by saying “Hello <whatever the name is>, I remember being <whatever the age is>. It was a great year!”

*Save as Conversation2*

## Project 5 – Selection 1 (IF…THEN…ELSE…Statements)

### The Maths Quiz Program

We can make our program react differently depending on what has been entered by the user. We will do this by comparing what the user has entered with what we are expecting. If they match, we can do something, otherwise we can make it do something different.

Start a new program and type in this code:

1. INPUT “What is 5 + 5 “ ; answer
2. IF answer=10 then PRINT “That is Correct” else PRINT “That is not correct”
3. END

*Save as MathsQuiz*

Notice that our IF Statement has 3 parts:

**IF** *<check the answer>* **THEN** *<do this if the answer is correct>* **ELSE** *<do this if the answer is wrong>*

**Now Try these:**

1. Extend the program to ask 10 questions instead of 1

*Save as MathsQuiz2*

1. Extend your program so that instead of telling you whenever you get the answer right, it adds one to your score. At the end of the quiz it should tell you how many you got right. You will need to introduce a new variable called score for this (At the start of the program, you can use the line **score = 1**)

*Save as MathsQuiz3*

**Extension:**

You can do more than one thing with an IF statement. E.g.

IF answer = correct

THEN

Do this

And this

ELSE

Do this

And this

And this

END IF

<Now carry on with the rest of the program….>

Extend MathsQuiz3 so that after each question, it adds 1 to your score AND tells you whether you got the answer right or wrong

*Save as MathsQuiz4*

## Project 6 – Repetition 1 (DO….LOOP)

### The guess the number program

So far, we have made programs that run once, and then end. We can make our program continue to run indefinitely until we tell it to stop. We will use a DO LOOP to do this. The DO LOOP lets us do something until we want it to stop. We will use this to create a number guessing game.

Start a new program and type in this code:

1. number=37
2. PRINT "In this program you will have to guess the number I am thinking of"
3. PRINT "It is a number between 1 and 100"
4. DO UNTIL guess=number
5. INPUT "Guess the number: "; guess
6. IF guess >number then print "That is too high"
7. IF guess < number then print "That is too low"
8. IF guess=number then print "That is correct"
9. LOOP
10. PRINT "You finished the game"

*Save as GuessingGame*

In this example, line 4 is the start of the loop. It tells the computer to keep repeating everything between line 4 and line 9 (the LOOP command) until guess=number (the guess is correct)

**Questions:**

Explain why the PRINT lines (lines 2,3 and 10) do not get repeated in this program

**Now Try these:**

1. Extend this program so that if keeps track of how many guesses you have made, and then tells you at the end of the game

*Save as GuessingGame2*

1. Extend MathsQuiz3 (or 4 if you managed to do it) so that it will ask you if you want another go when you have finished. **You could make it so that you enter 1 to play again, or 2 to finish**

*Save as MathsQuiz5*

**Extension:**

You can create a random number between 1 and 100 by using the INT and RND Functions:

**INT(RND(1)\*100)**

Use this to improve GuessingGame2 so that you can keep playing and there is a different, random number to guess each time. Save as GuessingGame3

## End of Unit Assessment:

**Create a program that simulates a game of snakes and ladders. Have a look on the internet to find what the board should look like. When the game starts, the computer asks your name. Then it asks you to roll the dice and enter your score. The computer should then work out what square you are on. If you are on a snake or a ladder square, the computer should take this into account. The game ends when you reach the last square (an exact throw is not required). When the game is finished, the computer should say congratulations and say how many dice rolls it took to finish the game.**

*Save as SnakesAndLadders*

When you have the basic program working, add some code to improve the feel for the game. You could add comments when a ladder or a snake is reached, or you could add a random dice roll into the game. You could even add an option to play again when the game is finished, or make it so you need to roll an exact number to finish.

*Save as SnakesAndLadders2*

# Learning Log

Use these pages to tick off the exercises as you complete them, and make notes on how you got on.

***For each program describe the problems you faced and what you have learnt***

## Section 1 (PRINT Command)

* HelloWorld
* MyName
* Twinkle

## Section 2 (Inputs)

* HowOldAreYou
* FaveNum
* FaveNum2

## Section 3 (Calculations and Operators)

* VATCalculator
* VATCalculator2
* RoomArea
* SheepPen
* TurfCalculator

## Section 4 (Strings)

* HelloYou
* Conversation1
* Birthday
* Conversation2

## Section 5 (IF, THEN, ELSE)

* MathsQuiz
* MathsQuiz2
* MathsQuiz3
* MathsQuiz4

## Section 6 (DO LOOP)

* GuessingGame
* GuessingGame2
* MathsQuiz5

## Final Assessment (Snakes and Ladders)

* SnakesandLadders
* SnakesandLadders2

*Congratulations!*

*You have now reached the end of the beginners* ***JustBASIC*** *Programming course*