# Module 1: Computer hardware and software

1.1 Different types of hardware of a common system

After you have completed this module, you should be able to :

* Identify the components of the system unit
* Explain the term CPU and its purpose
* Describe the term CPU and explain the impact of using various different types of CPU’s
* Describe different types of CPUs with regard to use, and power
* Define the term computer memory
* Discuss the primary purpose of memory
* Differentiate between different types of memory and their purpose (Range: RAM, ROM, CMOS, Cache memory, Flash Memory)
* Explain how data is stored on memory
* Define the purpose of the motherboard and its components
* Describe different types of input hardware with regard to use and classification e.g. direct and indirect entry
* Discuss how data is transferred between memory i.e. primary and secondary and the CPU
* Describe the Flow/transfer of data between components. (Range: USB – PnP, U3, Point-to-point connections)
* Describe the factors to consider when choosing an input device. (Range: Ergonomic considerations, Wireless vs cables)
* Describe different types of output hardware
* Describe the purpose and use of devices such as docking stations for mobile and laptop computers.
* Describe the term software
* Describe the purpose and function of software
* Describe the basic concepts of software. (Range: Software as programs. Identify software components. Concept of a graphical user interface (GUI).)
* Contrast: System software vs application software
* Differentiate between: Shareware, Freeware, Open Source Software and Proprietary software, Firmware
* Discuss the process of how software is obtained and installed.
* Differentiate between online software and installed software.
* Discuss the following terms in relation to software. (Range: Compatibility issues, Versions, patches and service packs, Updating software)
* Launch a new Linux terminal on the Raspberry Pi
* Use the man command to get help
* Expand a Linux file path and explain each element
* List the contents of the current folder using the ls command
* Change directly location using the cd command
* Create a new folder using the mkdir command
* Remove a folder using the rmdir command
* Remove a file using the rm command
* Rename a file using the mv command
* Copy a file using the cp command
* Clear the command prompts screen using the cls command
* Run an executable file from the command line

# FORMATIVE ASSESSMENT 1.1 INDIVIDUAL TASK

1.1.1 A system unit is the part of a computer that houses the primary devices that perform operations and produce results for complex calculations. ✓ ✓ (2)

1.1.2 The purpose of the CPU is to carry out the set of instructions given to the processor from a program. ✓ ✓ (2)

1.1.3 Information Processing Cycle Stages

1. Fetch- Instruction from Memory (Instruction Fetch, IF) ✓
2. Decode the instructions into binary (Instruction Decode, ID) ✓
3. Execute action and move to next step or calculate address (EXE) ✓
4. Access memory operand (MEM)✓
5. Write back result to register (WB) ✓ (5)

1.1.4 The rate at which one operation is completed in a second is measured in **hertz**. ✓ (1)

1.1.5 THREE main components of the CPU.

* Control Unit✓
* Arithmetic Logic Unit✓
* Registers✓ (3)

1.1.6 Differences between RISC and CISC

* In RISC, the instruction set is reduced, and most of these instructions are very primitive, while in CISC, the instruction set is very large that can be used for complex operations. ✓
* RISC computer’s execution time is very less, whereas CISC computer’s execution time is very high. ✓
* In RISC, the decoding of instructions is simple, whereas, in CISC, the decoding of instructions is complex. ✓ (6)

1.1.7 FOUR functions of a computer

1. **Data processing-** Data can take many forms, and the processing requirements are numerous.
2. **Data storage-** Even if the computer is processing data on the fly, it must temporarily store at least those pieces of data that are currently being worked on.
3. **Data movement-** The operating environment of a computer is made up of devices that act as data sources or destinations.
4. **Control-** A control unit manages the computer's resources and orchestrates the performance of its functional parts in response to commands.

1.1.8 The memory hierarchy is an improvement of computer storage into a hierarchy-based modal on response time. ✓✓ (2)

1.1.9 Static Random Access Memory and Dynamic Random Access Memory. ✓✓ (2)

1.1.10 Secondary memory is computer memory that is non-volatile and persistent in nature and is not directly accessed by a computer/processor. ✓✓ (2)

1.1.11 Hardware Devices

|  |  |  |
| --- | --- | --- |
| **Input devices** | **Processing devices** | **Output Devices** |
| Mouse✓ | Motherboard✓ | Monitor/Screen✓ |
| Keyboard✓ | Arithmetic Logic Unit✓ | Printer✓ |
| Light pen✓ | Processor✓ | Plotter✓ |

**(8)**

**Total:42 Marks**

# FORMATIVE ASSESSMENT 2.2 INDIVIDUAL TASK

1.2.1 Software – set of instructions, data or programs used to operate computers and execute specific tasks. ✓✓ (2)

1.2.2 THREE types of language translators (3)

Translators✓

Compilers✓

Assemblers✓

1.2.3 Compiler-A compiler is a computer program that transforms whole code written in a high-level programming language at once into the machine code✓✓ whereas an interpreter is a computer program, which converts each high-level program statement into the machine code line by line. ✓✓ (4)

1.2.4

**Freeware-** Freeware is the software that is available to use for free of cost without any limitations. ✓✓

**Middleware**- is software that is used to bridge the gap between applications and other tools or databases. Some examples of middleware activities include handling data and [API](https://www.techtarget.com/searchapparchitecture/definition/application-program-interface-API) management, authentication and messaging services. ✓✓

**Shareware**- The software is copyrighted and distributed for free only for testing purposes. After the trial period ends, you must pay. ✓✓

**Open source-** This is provided for use, modification, and redistribution. Open-source software is downloaded from the internet at no cost. ✓✓

(8)

1.2.5 **FOUR** categories of system software.

* Operating System✓
* Device Driver✓
* Language Translator✓
* Utility Programs✓

1.2.6 Benefits of online software

1. No software to install- Software is installed from the cloud. Once users have a license, it is easy to install. In addition, you can always retrieve the licence if lost. ✓✓
2. Complete flexibility on device type-Users do not need to be very careful with regards to storing the software. ✓✓
3. No maintenance required. - The responsibility is taken care of by the providers of the software or those who manage the cloud storage. ✓✓
4. Connect from anywhere at any time with internet- users can install the software from any geographical location and any time of the day. ✓✓
5. Centralised storage offering safe and secure data✓✓ (10)

1.2.7 Compatibility is the capacity for two systems to work together without having to be altered to do so. ✓✓ (2)

**Total:29 Marks**

# FORMATIVE ASSESSMENT 1.3 INDIVIDUAL TASK

1.3.1 To launch Linux terminal on Raspberry Pi, you need to do the following steps:

Step 1: From the Raspberry Pi icon✓, select Accessories✓

Step 2: From the Accessories, click terminal✓ (3)

1.3.2 ls- is used to list all files and directories in a file path on the Raspberry terminal. (1)

1.3.3 Raspberry Pi terminal commands

pi@raspberrypi:~/$cd Desktop✓

pi@raspberrypi:~Desktop $ mkdir terminal\_commands✓

pi@raspberrypi:~/Desktop $ cd terminal\_commands✓

pi@raspberrypi:~/Desktop/terminal\_commands $ pwd✓✓

/home/pi/Desktop/terminal\_commands

pi@raspberrypi:~/Desktop/terminal\_commands $

pi@raspberrypi:~/Desktop/terminal\_commands $ touch summative.docx✓

(6)

1.3.4 Adding text into the file called summative.docx

pi@raspberrypi:~/Desktop/terminal\_commands $echo “Hello fellow programmers”>> summative.docx✓✓✓

(3)

1.3.5 Merging contents of files.

pi@raspberrypi:~/Desktop/terminal\_commands $

pi@raspberrypi:~/Desktop/terminal\_commands$touch formative.docx

$ echo "Terminal commands are areasy">>formative.docx

$ cat formative.docx summative.docx >> final.docx

$ tac final.docx

(8)

1.3.6 Pairing the commands to the use

|  |  |
| --- | --- |
| **Command** | **Function** |
| rm | This command is used to delete files within a directory. |
| rmdir | To permanently delete an empty directory. |
| man | Is used to know more about a command and how to use it |
| mkdir | Used to create a new directory |

**(4)**

**Total: 25 Marks**

# SUMMATIVE ASSESSMENT 1.4 INDIVIDUAL TASK

1.4.1 The difference is that a Memory buffer register (MBR) contains a word to be stored in memory or sent to the I/O unit or is used to receive a word from memory or from the I/O unit. ✓✓whereas Memory address register (MAR) specifies the address in memory of the word to be written from or read into the MBR. ✓✓ (4)

1.4.2SRAM: is a memory chip that is faster and uses less power than DRAM✓✓.whereas DRAM is a memory chip that can hold more data than an SRAM chip, but it requires more power. ✓✓ (4)

1.4.3 FIVE factors to consider when choosing an input device.

1. **User Needs**- this is the urgency of use of the device in the computer room by users also whether it will satisfy the needs of the user. ✓✓
2. **Initial cost-**the amount it can cost when buying/purchasing the devices✓✓
3. **Maintenance Cost**-the amount that can be used to maintain the servicing of these devises should be considered. ✓✓
4. **Mode Of Transmission**- how will you transport them to the computer room should be considered✓✓
5. **Compatibility With Available Hardware**- will the devices fit to other devices in the room already. A device that fits your needs but doesn't work with your computer is useless. ✓✓
6. **User- Friendliness**- whether the devices will be used to solve problems and easy to be used by users✓✓ (10)

1.4.4 Types of ROM

1. **MROM (Masked Read Only Memory)** ✓
2. **PROM (Programmable Read Only Memory)** ✓
3. **EPROM (Erasable and Programmable Read Only Memory)** ✓
4. **EEPROM (Electrically Erasable and Programmable Read Only Memory)** ✓

**(4)**

**1.4.5 Differences between microcontroller and microprocessor**

1. **Microprocessors only have a Central Processing Unit, whereas Micro Controllers have a CPU, Memory, and I/O all integrated into a single chip.** ✓✓
2. **Microprocessors are used mainly in personal computers, whereas microcontrollers are used in embedded systems.** ✓✓
3. **Microprocessors are based on the Von Neumann model while microcontrollers are based on the Harvard architecture.** ✓✓ (6)

1.4.6 Proprietary software is computer software where the source codes are publicly not available only the company that has created can modify it. ✓✓ (2)

1.4.7 Forward compatibility is a design principle in which a program or piece of hardware is designed to work with new software or devices in the future. ✓ (1)

1.4.8 False (1)

1.4.9 The terminal will print the message in quotation marks and repeat the process id twice. Example: The process id is 590590✓✓ (2)

1.4.10 Working with terminal commands

pi@raspberrypi:~/$cd Desktop✓

pi@raspberrypi:~Desktop $ mkdir Practical✓

pi@raspberrypi:~/$cd Practical✓

pi@raspberrypi:~Desktop/Practical $ echo "It is the control unit that determines which machine instruction is to be executed next" >> next\_file.txt✓

pi@raspberrypi:~Desktop/Practical $ echo " The control unit is the brains of the computer " >> next\_file.txt✓

pi@raspberrypi:~Desktop/Practical $ pwd✓

(6)

**Total : 40 marks**

# PRACTICAL ACTIVITY PAIR WORK

**Software Installation**

Search for Anydesk and install the software on your machine. You can go to this link: <https://anydesk.com/en/downloads/thank-you?dv=win_exe>

Share the pin numbers of the software with your friend and try controlling each other’s machine.

The software allows remote control of one’s PC. This is quite helpful if you are working with a friend or classmate and he/she gets stuck but you are in a different geographical location. Also, if your friend or colleague is not able to follow the instructions, you can take full control of his or her computer and start assisting remotely.

# Module 2 Problem solving in computer programming

After you have completed this module, you should be able to :

* Define the term problem solving
* Define the term computational thinking
* Describe the phases of the PLDC (Program Development Life Cycle)
* Describe the purpose of problem solving leading to solutions
* Explain and apply various problem-solving steps.
* State in own words Clarity on what needs to be done
* What is known or given?
* What is missing or needed?
* Devise a plan/algorithm (storyboard – visual or
* textual)
* Look for patterns
* Look at related problems, known solutions
* Examine simpler or special cases
* Make a table, create diagram, use guess and check,
* work backwards, identify sub-goal
* Carry out the plan/implement the algorithm (write
* the code)
* Look back/test (see if it works)
* Check results against original problem. Does it make sense? Is there another solution?)
* Use appropriate tools and techniques to present a solution. Range:
* User stories (written by the client and provide the requirements)
* Noun-verb analysis of user stories
* List of nouns provides identification of objects and state
* List of verbs provides identification of behaviour
* Acceptance tests (does the program meet the
* requirements?)
* Define the term algorithm and its purpose in the problem-solving process. (Range: Basic concepts of an algorithm.
* What is an algorithm? Develop a clear understanding of the problem presented.)
* Implement and understand the basic algorithmic constructs used to create a flowchart. Range: Input, Output, Processing and Calculations, Selection, Iteration
* Create a flowchart to present a particular algorithm and its associated tasks
* Interpret a basic flow chart and describe its intended operation / function

# FORMATIVE ASSESSMENT 2.1 INDIVIDUAL TASK

2.1.1 Problem solving is the sequential process of analysing information related to a given situation and generating appropriate response options.✓ ✓ (2)

2.1.2 Computational thinking is an interrelated set of skills and practices for solving complex problems, a way to learn topics in many disciplines, and a necessity for fully participating in a computational world. .✓ ✓ (2)

2.1.3 **FOUR** cornerstones of computational thinking are:

* **decomposition** ✓
* **pattern recognition** ✓
* **abstraction** ✓\P
* **algorithms** ✓ (4)

2.1.4 Program Development Lifecyle

(6)

2.1.7 SIX steps which must followed in problem solving.

* 1. Understand the Problem ✓
  2. Formulate a Model ✓
  3. Develop an Algorithm ✓
  4. Write the Program ✓
  5. Test the Program ✓
  6. Evaluate the Solution✓

(6)

2.1.8 Write an algorithm that reads three numbers and prints the value of the largest number. (10)

2.1.8 Algorithm

Step1: Start

Step2: Read/input A,B and C

Step3: If (A>=B) and (A>=C) then Max=A

Step4: If (B>=A) and (B>=C) then Max=B

Step5:If (C>=A) and (C>=B) then Max=C

Step6: Print Max

Step7: End

(10)

2.1.9 Algorithm is the sequence of steps to be performed in order to solve a problem by the computer. ✓✓ (2)

**Total 32 Marks**

# FORMATIVE ASSESSMENT 2.2 INDIVIDUAL TASK

2.2.1 FOUR testing phases stages of program development lifecycle

*Software Testing* is a method to check whether the actual software product matches expected requirements and to ensure that software product is[defect](https://www.guru99.com/defect-management-process.html)free.

* **Integration testing**- The goal of integration testing is to test the interfaces between modules and expose any defects that may arise when these components are integrated and must interact with one another. ✓✓
* **Unit testing**- Unit testing is typically performed throughout the application development process, with the goal of ensuring that every single unit or component works as expected. ✓ ✓
* **Acceptance Testing**- It is a type of testing done to ensure that the users' requirements are met prior to delivery and that the software works properly in the user's working environment. ✓✓
* **System Testing :** **Complete and integrated software’s are tested in system testing, which means that all of the system elements forming the system are tested as a whole to meet the system's requirements.** ✓✓
* **Acceptance testing** is a test used to determine whether or not the requirements of a specification or contract have been met. ✓✓
* **Regression testing**- This determines if adding additional features results in a decrease in an application's functionality. ✓✓

2.2.2 Flowchart



(10)

2.2.3 SIX characteristics of a good algorithm.

* **Unambiguous**✓
* **Input-must have 0 or more inputs**✓
* **Output** − should have 1 or more well-defined outputs. ✓
* **Finiteness**-must terminate after several steps✓
* **Feasibility**-should be feasible with available resources✓
* **Independent**-must have step by step directions independent from other programs✓

2.2.4 Average of 25 scores.

A picture containing diagram

Description automatically generated

2.2.5 Algorithm to calculate area of Circle.

**Step1**: Start

**Step 2**:Input the Radius r of the Circle

**Step3**: Area PI\*r\*r // calculation of area

**Step4**: Print Area

**Step 5**: Stop

(6)

2.2.6 SIX benefits of using flowcharts.

* **Communication:** Flowcharts are a better way to communicate the logic of a system to all parties involved. ✓✓
* **Effective analysis:** A flowchart can help you analyse a problem more effectively. ✓✓
* **Proper documentation:** Program flowcharts are useful for program documentation, which is required for a variety of reasons. ✓✓
* **Efficient Coding:** During the systems analysis and program development phases, the flowcharts serve as a guide or blueprint. ✓✓
* **Proper Debugging:** The flowchart aids in the debugging process. ✓✓
* **Efficient Program Maintenance:** The use of a flowchart simplifies program maintenance. It allows the programmer to focus his or her efforts more effectively on that aspect. ✓✓ (12)

**Total : 66 Marks**

# SUMMATIVE ACTIVITY 2.3 INDIVIDUAL TASK

2.3.1 **FOUR** cornerstones of computational thinking are:

* **decomposition** - breaking down a complex problem or system into smaller, more manageable parts
* **pattern recognition** –similarities among and within problems
* **abstraction** – focusing on the important information only, ignoring irrelevant detail
* **algorithms** - developing a step-by-step solution to the problem, or the rules to follow to solve the problem (8)

2.3.2 A conditional flowchart is a design technique used when a condition is imposed on a problem. (2)

2.3.3 Flowchart for Fahrenheit to Celsius

Graphical user interface, application

Description automatically generated

**(5)**

2.3.4 Quadratic equation

Algorithm (7)

Step 1: Start

Step 2: Input a, b, c

Step 3: Calculate d. d = sqrt(b x b – 4 x a x c

Step 4: Calculate x1. x1 = (–b + d) / (2 x a)

Step 5: Calculate x2. x2 =(–b – d) / (2 x a)

Step 6: Print x1 and x2

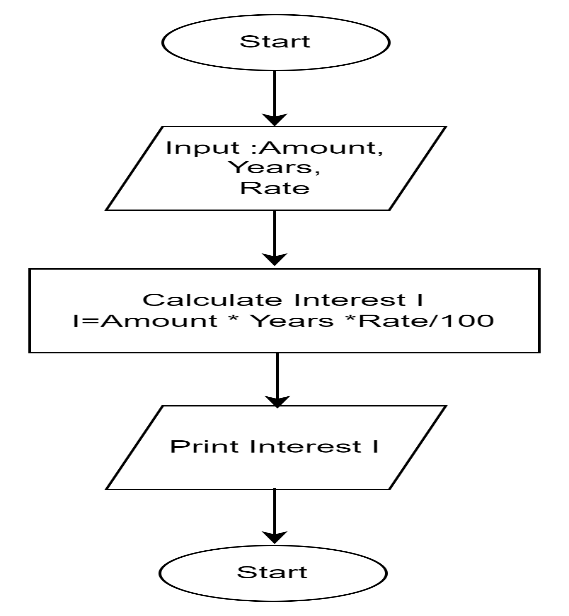
Step 7: Stop

Flowchart (7)

Qr code

Description automatically generated with medium confidence

2.3.5 Computing Interest. **I=Amount \* Years \*Rate/100**

 (7)

2.2.6 THREE limitations of flowcharts

* **Complex logic:** The program logic can be quite complicated at times and in such instances, flowchart become more complex and clumsier.
* **Alterations and Modifications:** If changes are required, the flowchart may need to be completely redrawn.
* **Reproduction:** Because flowchart symbols cannot be typed, reproduction of flowcharts is difficult. (6)

* + 1. The symbol denotes \_\_\_\_\_\_\_

1. I/O
2. Flow
3. **Process**
4. Decision

2.2.8 A box that can represent two different conditions.  
a) Rectangle  
**b) Diamond**  
c) Circle  
d) Parallelogram

2.2.9 In computer science, algorithm refers to a pictorial representation of a flowchart.  
a) True  
**b) False**

2.2.10 The operation represented by parallelograms is \_\_\_\_\_\_\_\_.  
**a) Input/Output**  
b) Assignment  
c) Comparison  
d) Conditions

**Total :46 Marks**

# Module 4 Building and running C/C++ applications

# FORMATIVE ASSESSMENT 4.1 INDIVIDUAL TASK

4.1.1 A compiler is a language processor that reads a whole source program written in high- level language in one go and converts it into an equivalent program written in machine code. (2)

4.1.2 Compiled languages.

* C, C++
* C#
* Java
* Erlang
* Haskell
* Rust
* Go.

**ANY OF THE THREE (3)**

4.1.3 The back end of compiler includes those portions that depend on the target machine and generally those portions do not depend on the source language, just the intermediate language. These include

1. Code optimization

2. Code generation, along with error handling and symbol- table operations.

(3)

4.1.4 Advantages of a compiler

**Advantages of compilers**

1. **Independence**-No other program or application is required to run the executable file of source codes.
2. **Optimisation**-The compiled program is well optimized and runs faster.
3. **Data Security**-The compiler generates executable files that can be executed on any other system.
4. **Speed-**Compiler are faster than interpreter

**Disadvantages of Compilers**

1. **Hardware Specific**
2. **Time consuming-** Compilation process takes time before an executable file is produces
3. **Extra memory**-Because source code object code file and executable files
4. **Debugging Difficulty**- All errors are shown at once.

(6)

4.1.5 An interpreter program executes other programs directly, running through program code and executing it line-by-line. (2)

4.1.6 Differences between compiler and interpreter.

| **Compiler** | **Interpreter** |
| --- | --- |
| A compiler translates the entire source code in a single run. | An interpreter translates the entire source code line by line. |
| Compilers are faster than an interpreter. | Interpreters are slower than the compiler. |
| CPU utilization is more. | CPU utilization is less as compared to the compiler. |
| Both syntactic and semantic errors can be checked simultaneously. | Only syntactic errors are checked. |
| The compiler is larger than interpreter therefore require more memory. | Interpreters are often smaller than compilers. |
| The localization of errors is difficult. | The localization of error is easier than the compiler. |

(12)

**Total : 28 Marks**

# FORMATIVE ASSESSMENT 4.2 GROUP TASK

**4..2.1** As part of software installation, you are expected to be able to install any IDE and the C++ compiler. We have listed a lot of IDE’s such as Code::Blocks, Geany

Task.

In groups, you are required to uninstall CodeBlocks and Mingw.

Reinstall Code::Blocks and Mingw

Test if gcc/g++ are available.

Run a simple hello world program.

(20)

# FORMATIVE ASSESSMENT 4.3 INDIVIDUAL TASK

4.3.1 GCC stands for “GNU Compiler Collection”. GCC is an integrated distribution of compilers for several major programming languages such as C, C++, Fortran, Ada, D, and Go. (2)

4.3.2 GCC stands for GNU Compiler Collection -on open-source collection of compilers for various languages based on a common core compiler and linker whereas MinGW stands for Minimal GNU for Windows. This is a copy GCC plus other common utilities and libraries used in compiling and linking C or C++ code using the GNU tools.

4.3.3 An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. (2)

4.3.4 Installing Geany on a Raspberry Pi.

Step 1: Navigate to the Raspberry Icon

Step 2: Select accessories and select terminal.

Step 3: Type the following command:

sudo apt-get install geany

Step 4: Press enter. Geany IDE will be installed.

(4)

4.3.5 Complete the following table to differentiate G++ from GCC. (8)

|  |  |  |
| --- | --- | --- |
| **Parameter** | **G++** | **GCC** |
| **Uses** |  |  |
| **Compilation** |  |  |
| **Compilation Command** |  |  |
| **File Linking in Library** |  |  |

4.3.5 Differences between G++ and GCC

|  |  |  |
| --- | --- | --- |
| **Parameter** | **G++** | **GCC** |
| Uses | G++ is used for compiling C++ | Used for compiling C |
| Compilation | can compile either.cpp or.c files, but they will only be treated and released as C++ files. | can compile either.cpp or.c files, but they will be treated and released as C++ or C |
| Command for Compilation | g++ fileName.cpp -o binary. | gcc fileName.c -o binary. |
| File Linking in Library | When we use the G++ command to link the object files, the files automatically link in the standard C++ libraries. | It does not happen in the case of GCC. |

(8)

# FORMATIVE ASSESSMENT 4.4 INDIVIDUAL TASK

4.4.1

a) Syntax errors- is a bug that occurs when a computer program has an incorrectly typed statement.

b) Runtime errors-is a type of error which during runtime and could be caused by inputting wrong data type.

c) Logic errors-occurs when program runs but gives incorrect results. (6)

4.4.2 FOUR reasons for debugging computer programs.

a) Debugging ensures that the final product is bug free.

b) Competitive advantage-Bug free programs will build a good reputation for the software firm.

c) Saves Time**-** Performing debugging at the initial stage saves the [time of software developers](https://www.educba.com/career-as-a-software-developers/).

d) Debugging ensures that requirements are met. (4)

4.4.3 Remote debugging technique. (1)

4.4.4 Stepping is a debugging technique that involves executing code one instruction or line at a time. (2)

4.4.5 Setting up a breakpoint

Text

Description automatically generated

Figure 4. 1: Sample code

Output.

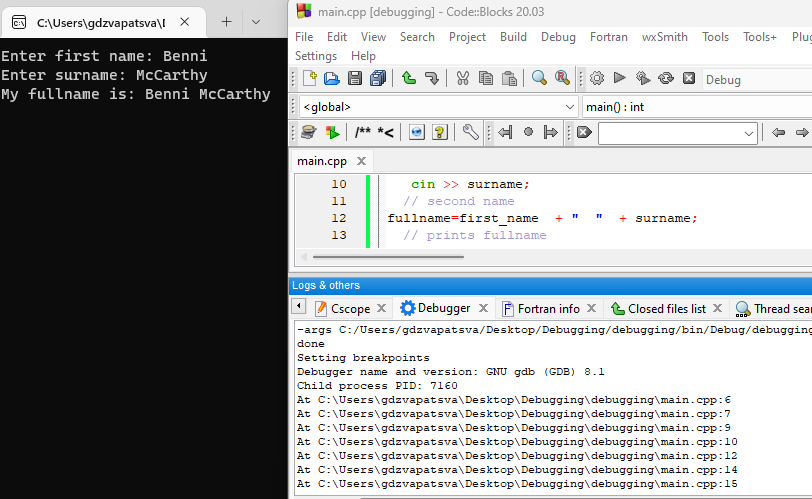


Figure 4. 2:Output

(10)

**Total: 22 marks**

# FORMATIVE ASSESSMENT 4.6 INDIVIDUAL TASK

4.6.1 Define the term version control. (2)

4.6.1 Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. (2)

4.6.2 What is the difference between Git and GitHub? (4)

4.6.2 Git is a free and open-source distributed version control software designed to handle everything from small to very large projects with speed and efficiency whereas GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git.

(4)

4.6.3 List FIVE benefits come with using GIT? (5)

4.6.3 FIVE benefits of using Git.

1. Data replication and redundancy are both possible.
2. It is a service with high availability.
3. There can only be one Git directory per repository.
4. Excellent network and disc performance are achieved.
5. On any project, collaboration is very simple.

4.6.4 Write the command used to initialise a Git repository. (2)

4.6.4 $ git init (2)

4.6.5 What does the command git -v do when typed on the git bash terminal. (2)

4.6.5 Git -v or git –version or git version gives the version of git software installed on the machine (pc or laptop). (2)

4.6.6