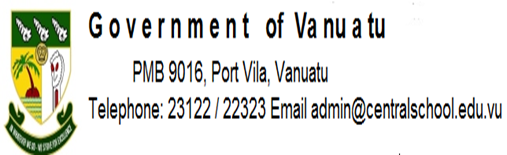
****

Central School

Home School Package

**Year : 13 ICT**



**HOME SCHOOL PACKAGE CONTENT**

Website Development

Programming

Microprocessor

**LESSON Plan**

|  |  |
| --- | --- |
| G:\Home Learning Packages\Documents for SHEFA Schools Principal\teacher-computer-icons-school-test-education-teaching.jpg Teacher | Name : Rensy Ilaisa  Subject : ICT |
| G:\Home Learning Packages\Documents for SHEFA Schools Principal\download.jpg  Date | May 25 – June 6 2020 |
| G:\Home Learning Packages\Documents for SHEFA Schools Principal\title.jpg | Topic : Website Development |
| Learning outcomesLearning outcomes | * Define website * Outline the principles of good website design. * Explain the importance of using the principles of good website design. * Define CSS. * Define HTML. * Create a web-page to present a text object using HTML codes. |
| TopicIntroduction | **1. Web design concepts**  Definition of website: a group of web pages usually containing hyperlinks to each other and made available online by an individual, company, educational institution, government, or organization.  **Elements of Good Web design**  Creating a great website is not an easy task. Every year new tools and ideas appeared and changed the way we design or develop websites. But the basic elements remain to provide the basis for designing or creating a good website no matter what the requirements are. Some of the common website design elements are; |
| Catch | How to create an attractive webpage using the concepts learnt in class. |
| Learners notes 1  Learners notes | |  |  | | --- | --- | | **Principles** | **Definition** | | Simple | Putting too many elements on the page may lead to distracting visitors from the main purpose of your website. Simplicity always works in an effective web page design. | | Color Scheme in Web Design | The selection of an appropriate colour scheme that meets the company choice and pleasing to look at. | | Consistency | Ensure that fonts, sizes, headings, sub-headings, and button styles must be the same throughout the website. | | Typography & Readability | No matter how good your design is text still rules the website as it provides users the desired information, so consider using fonts that are easier to read. | | Usability | It’s very important to check that everything on the site is working as planned. The site is easy to navigate; quick load time and users do find a lot of useful resources and information. | | Mobile Compatibility | Keeping in mind the ever-growing usage of smartphones, tablets, and phablets, web design must be effective for various screens. | | Easy Loading | No one likes the website that takes too much time to load. So, take care of it by optimizing image sizes, combing code into a central CSS or JavaScript file as it reduces HTTP requests. | | Easy Navigation | For effective navigation, you may consider creating a logical page hierarchy, using bread scrums, and designing clickable buttons. | | Communication | The ultimate purpose of the visitors is to get information, and if your website is able to communicate your visitors efficiently, most probably they would spend more time on your website. | | Clear Contact Information | A good design must include easy access to contact information such as phone number or email address whenever and wherever possible. |   A good design provides the basis for a successful website. A nicely-designed website provides the face of a business, a school or an organization. Therefore, the design must be able to attract the target audience and customers to the site and to do that, a website designer MUST follow the principles of good website design. For a business, lacking in any aspect of the design could end up damagingits brand impression.  *“Web page design is more critical for conversions than you think. Despite using a great conversion boosting tactic, you may not do much if it looks poor in quality. In fact, website design doesn’t necessarily mean how it looks like and feels like but is how it works. Even a simple looking website with exceptional usability and well-structured typically performs amazingly on Google. User views of such websites are also higher than those with poor user experience. The performance entirely depends on the effectiveness of the website.”*  [*https://wpastra.com/good-website-design/*](https://wpastra.com/good-website-design/)  **Web-design requirement**  The World Wide Web (www) is not just a huge market place for business opportunities but it’s a place for sharing ideas, entertainment and social engagements.  Having a website is an important business decision and it’s easy and not so expensive. This website can provide more information why it’s so important to have one;  [*https://bigmouthmarketing.co/why-do-i-need-a-website/*](https://bigmouthmarketing.co/why-do-i-need-a-website/)  Some of the main web-design requirements are;   * Company brand such as the company logo, preferred colour schemes, specific design templates, etc * Basic features & functionality such as e-commerce capabilities; support multiple languages; have a business directory listing or mobile version. * Company information or content to be shared online * A proper layout structure of how the company information is going to be * Type of people you want access to your website and their interest * Look and feel such as graphics, required menu and design elements (borders, buttons, headers, etc) * Security required for protecting the website files and company information * Database if dynamic information is used so required forms for database access will be required * Type of web design tools or software to be used   **Web-design tool or software**  Building a website today is so much easier than it used to be 5 or 10 years ago. With newer web-design software that’s widely available today, anyone with some computer experience can build a good website at a very little time and cost.  Most web design software today are;   * Fairly easy to use with drag & drop capabilities and no coding required. * Professionally designed templates to use. * All technical maintenance work is taken care of for you. * 24/7 dedicated support teams. * Affordable pricing.   Some simple web design software can be accessed online such as;  Wix, Wordpress, Joomla  **Importance of the requirements**  You may know how well to design a website but failing to fulfil the requirements for developing a website can lead to failure as well. Therefore, knowing the requirements and plan well before designing a website is very important.  For a business, a website is used as a form of advertising and it allows potential clients to learn more about the business and the available products it has to offer. There are certain things that a website must have to ensure that the website meets its main purpose; and that is to help the business sells its products and attract more and more clients.  Planning for a website involves setting a goal for the website to ensure that the design is well guided. Some examples to consider as a goal for a website design are;   * Increase brand awareness * Capture emails or social followers * Generate more sales and revenue * Get more phone calls to your business * Sell products and services   Another important part of the plan is to know the target audiences. An audience’s behavior & interests will impact the structure of the website as well as the design elements to be included on the website.  It is also advisable to carry out a proper review of similar websites as part of the plan in order to identify key strengths and weaknesses of these websites that can help with preparing a new design.  **2. Web Design**  **CSS**  CSS stands for Cascading Style Sheets. CSS are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.  **HTML**  HTML stands for HyperText Markup Language. HTML is the standard language used to creating web pages.  HTML defines the structure and layout of a Web document by using a variety of tags and attributes. The correct structure for an HTML document starts with <HTML><HEAD>(enter here what document is about)<BODY> and ends with </BODY></HTML>. All the information you'd like to include in your Web page fits in between the <BODY> and </BODY> tags.  **CSS & HTML**  HTML and CSS are two different types of markup (code), which have their own unique syntax (the arrangement in which code is written). HTML defines the structure for the web page, while the CSS gives the HTML it’s styling.  HTML = structure; CSS = style  CSS provides web designers with the ability to create common style sheets. HTML provides the structure for each web page. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Under the HTML page, a reference to a CSS file is defined linking the CSS files and the HTML file. CSS saves a lot of work by controlling the layout of multiple web pages all at once. Both the HTML file and CSS file must be in the same web folder.  Example:  <!DOCTYPE html>  <html>  <head>    <title>This is my page title.</title>    <link href="style.css" rel="stylesheet" type="text/css" />  </head>  <body>    <h1>This is a heading 1 element</h1>    <p>Hello world, this is a simple paragraph.</p>  </body>  <html>  The web page created above is linked to an external css file named: style.css  h1 {  color: blue;  }  **3. Testing and Validating a website**  Web Testing in simple terms is checking your web application for potential bugs before it’s made live.During this stage issues such as that of web application security, the functioning of the site, its access to handicapped as well as regular users and its ability to handle traffic is checked.  A basic test is a using a web browser to check the functionality of the website. This is checking if the website and all the links are working and meet the specifications that the site was intended for.  Test all links in your webpages are working correctly and make sure there are no broken links.  Validating a website is the process of ensuring that the pages on the website conform to the norms or standards defined by various organizations. Validation is important and will ensure that all web pages are interpreted in the same way (the way preferred by the designer) by various machines, such as search engines, as well as users and visitors to the website.  The W3 Consortium has its own online validator which you can use for free. It may be found at: <https://validator.w3.org/>  HTML validator checks to make sure the HTML code on your web page complies with the standards set by the W3 Consortium. A CSS validator checks your Cascading Style Sheet in the same manner. That is, it will check that it complies with the CSS standards set by the W3 Consortium. |
|  | <https://www.youtube.com/watch?v=alswD2tCc_Q>  <https://www.youtube.com/watch?v=3JluqTojuME> |
|  | 1. Define website. (L1) (inf2.1.1.1) 2. Fill the table with some of the key principles of good website design. 3. A single principle. (L1) (inf2.1.1.2) 4. Multiple principles. (L2) (inf2.1.2.1)  |  | | --- | | **Principles of good website design** | |  | |  | |  |  1. Explain the importance of using the principles of good website design stated above. (L3) (inf2.1.3.1) 2. Discuss the importance of key web design requirements for an organisation. 3. A key web design requirement provided. (L1) (inf2.1.1.3) 4. An outline of key website design requirements. (L2) (inf2.1.2.2) 5. A full explanation of the importance of key website design requirements. (L3) (inf2.1.3.2) 6. A complete discussion of the importane of key website design requirements. (L4) (inf2.1.4.1) 7. Fill the table below with the definitions of the following web desing terms;   (L1) (inf2.1.1.4) (inf2.1.1.5)   |  |  | | --- | --- | | **Terms** | **Definition** | | CSS |  | | HTML |  |  1. Explain the advantages of using CSS in developing websites. 2. An advantage of using css. (L1) (inf2.1.1.6) 3. An outline of the advantages of using css. (L2) (inf2.1.2.3) 4. An explanation of the advantages. (L3) (inf2.1.3.3) 5. Create webpages by following the instructions below. 6. A single web-page. (L1) (inf2.1.1.7) 7. Mulitpleweb-pages. (L2) (inf2.1.2.4)   i. Login to your computer and create a folder on the desktop “My web files”  ii. Run the text processor “Notepad”  iii. Copy the following HTML tags to the Notepad window.  <!DOCTYPE html>  <html>  <body>  <h1>My First Heading</h1>  <p>My first paragraph.</p>  </body>  </html>  **Note:**   * You can change the “My First Heading” to your own Title for example “My first website” * You can change the “My first paragraph” to include your own text for example “My first website ……..”   iv. Save the file to your web files folder using your **name** as the filename with “html” as the extension for example ***frank.html***.  v. Use your browser to open the html file.  vi. You can edit the html file using the following tags; Save your file after you made changes. Use the browser to check.  Bold - <b></b> for example to bold the title <title><b>Title of Page</b></title>  The text in between the tags will be bold, and stand out against text around it, the same as in a word processor.  Italic - <i></i> Also working the same way as a word processor, italics displays the text at a slight angle.  Underline - <u></u> Again, the same as underline in a word processor. Note that html links are already underlined and don't need the extra tag.  Font Colour - <font color="#??????"></font> (\*)  Change the colour of a few words or a section of text. The 6 question marks represent the hex color code. Use google to search for the html colour codes. Try using some colours on your page.  Font Size - <font size="?"></font>  Replace the ? with a number from 1 to 7 to change the size of the font. One being the smallest and seven the largest.   1. Create webpages with links to media files by following the instructions below. 2. Multiple web-pages with hyperlinks to media files. (L3) (inf2.1.3.4) 3. Mulitpleweb-pages with hyperlinks to media files to meet specific requirements (L4) (inf2.1.4.2)   i.Login to your computer and make sure that you created a folder “My web files” on the desktop to store your files (html file, graphic file, audio file and video file)  ii.Use your browser to download some media files from the Internet (graphic, audio and video files)  iii. Create and name 3 new folders inside your “My web files” folder and copy the Graphics, Audio and Video files to the appropriate folder (graphic files to the graphic folder and so on).  Name the files: graphics.jpg for the graphics file; video.mp4 for the video file and audio.mp3 for the audio file.    iv. Run Notepad and enter the following tags.  <!DOCTYPE html>  <html>  <body>  <h1>My First Heading</h1>  <p>My first paragraph.</p>  <h2>My graphic file</h2>  <imgsrc=" graphics/graphics.jpg " alt="graphics" style="width:304px;height:228px;">  <h2>My audio file</h2>  <audio controls>  <source src="audio/audio.mp3" type="audio/mpeg">  </audio>  <h2>My video file</h2>  <video width="320" height="240" controls>  <source src="video/video.mp4" type="video/mp4">  </video>  </body>  </html>  **Note:**  i. You can change the “Title of Page” to your own Title for example “My first website”  ii. You can change the “Text of page…” to include your own text for example “My first website ……..”  iii. Save the web file again and make sure it has the html extension.  iv. Use your browser to open the html file.   1. Create an external CSS file to control the style of the webpage created above. 2. A basic css file. (L1) (inf2.1.1.8) 3. A complete css file controlling the style of multiple web-pages. (L2) (inf2.1.2.5) 4. Use your browser to test your web-pages. 5. Able to test a web-page only. (L1) (inf2.1.1.9) 6. Able to test the entire website. (L2) (inf2.1.2.6) 7. Outline the advamtages of validating a website. 8. A single advantage. (L1) (inf2.1.1.10) 9. Multiple advantages. (L2) (inf2.1.2.7) 10. Outline the steps involved in the validation process of websites. 11. A key part of the process. (L1) (inf2.1.1.11) 12. An outline of the steps. (L2) (inf2.1.2.8) 13. Validate the website created above [both HTML and CSS]. 14. Validation of a single web-page successfully. (L1) (inf2.1.1.12) 15. Validation of the website successfully. (L2) (inf2.1.2.9) |
| Assignment | Do this activity as an assignment as i twill be assessed. |
| Assessment | . Create a webpage of at least 3 links. |
| Reference ClipartReferences |  |

**WEEKLY CHECKLIST For Parents**:

Term: 2 Week number 1-3 Date: May 25 to June 6 Month: June

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Subject** | **Number of lessons** | **Days** | **Tickwhenactivityiscomplete** | **Parents comment** | **Signature** |
| **ICT Website Development** | **1-5**  **Wk 1** | **Monday - Friday** |  |  |  |
| **ICT**  **Website Development** | **1-5**  **Wk2** | **Monday - Friday** |  |  |  |
| **ICT**  **Website Development** | **1-5**  **Wk3** | **Monday - Friday** |  |  |  |

|  |  |
| --- | --- |
| G:\Home Learning Packages\Documents for SHEFA Schools Principal\teacher-computer-icons-school-test-education-teaching.jpg Teacher | Name : Rensy Ilaisa  Subject : ICT |
| G:\Home Learning Packages\Documents for SHEFA Schools Principal\download.jpg  Date | June 15, 2020 to June 20, 2020 |
| G:\Home Learning Packages\Documents for SHEFA Schools Principal\title.jpg | Topic :Programming  Lesson number : 1-5 |
| Learning outcomesLearning outcomes | * Define problem-solving. * Identify a step in the problem-solving process. * Explain the problem-solving process. * Outline key benefits of using the problem-solving process. |
| TopicIntroduction |  |
| Catch | Problem solving concepts |
| Learners notes 1  Learners notes | **1. Problem solving concepts**  Problem solving is the source of all new inventions, social and cultural evolution, and the basis for market-based economies. It is the basis for continuous improvement, communication and learning.  Definition: The process of working through details of a problem to reach a solution.  Computer programming is like solving a problem. A program is a defined set of steps to solve a problem by;   * using code to represent the problems * define the steps required to solve the problem using algorithm  1. Identify the problem. 2. Break down the problem into smaller parts. 3. Identify solution for each part. 4. Write the steps for the solution using pseudo-code, algorithm and flow diagrams. 5. Replace pseudo-code or algorithm with real programing code. 6. Run the program to produce the required solution.   Creating a computer program is developing the code in order to automate the process required to solving a problem. But nothing can be automated that cannot be done manually. Any code is written is based on the steps that can be used to solve the problem manually. So, the problem-solving process is used to know exactly the steps that are required to be automated; this will save a lot of time wasted in trying to create the program by just starting with writing the code.  Problems are around us and we have different ways of dealing with them. Unless you have a problem-solving system, this is probably how you “solve” problems;   * Try a solution. * If that doesn’t work, try another one. * If that doesn’t work, repeat step 2 until a solution is found.   If you’re lucky you may find a solution, but this is the worst way to solve problems and it’s a huge, huge waste of time. In computer programming, waste of time can be very costly if you’re dealing with a programming project that requires quality and efficient output.  Well known programmers have stated that the biggest mistake that new programmers make is focusing on learning programming syntax (coding) instead of learning how to solve problems. Programming is providing solution to a problem but not just providing codes for a program.  To have a good system for finding the solution to a problem, one has to follow the following;   * Make an effort to fully understand the problem. * Prepare a plan by giving enough time to analyze the problem and process the information. * Do not try to solve one big problem but break it into sub-problems; these sub-problems are much easier to solve by starting with the simplest. * Go step by step through the solution to ensure that everything is OK or if there is something wrong.   Problem solving is a skill that requires practice and more practice. Once you master the skills, programming and coding will be easy and importantly, you can create better programs that meet their intended outcome (able to create program solutions).  Problem solving skills and the problem-solving process are a critical part of daily life both as individuals and organisations. Developing and refining these skills through training, practice and learning can provide the ability to solve problems more effectively.  Example 1:  i. Identify a problem that requires a computer program solution.  “Making a cup of coffee”  ii. Decompose the problem into solvable components.   * Fill the kettle with water * Boil water * Preparing the cup * Pouring of hot water and add sugar and milk   iii. Describe the solution to the major part of the problem in plain English.  Start with filling and heating a kettle of water. Once the water boils, then pour the hot water to a mug that has a tea spoon of coffee, sugar and milk. Use a spoon to stir so that you get a good mix before serving.  Example 2:  i. Identify a problem that requires a computer program solution.  “Finding the average of integers entered through the keyboard”  ii. Decompose the problem into solvable components.   * Entering of numbers: how many numbers * Finding the average: using a formula for finding the average * Printing the output   iii. Describe the solution to the major part of the problem in plain English.  Start with asking how many numbers to find the average and then ask for the numbers to be entered using the keyboard. Use the formula for finding the average and then print the average as the output. |
|  | <https://www.youtube.com/watch?v=lx7oqZ7Nl3k>  <https://www.youtube.com/watch?v=cybMiQYuEwc&list=PLH1n1sJO7tbxmE36txTPhgidmW5Z9Bn7m>  <https://3schools.com> |
|  | 1. Define problem-solving.(L1) (inf3.1.1.1) 2. Explain the problem-solving process. 3. A step in the problem-solving process was identified. (L1) (inf3.1.1.2) 4. An outline of the steps involved in the process. (L2) (inf3.1.2.1) 5. An explanation of the process. (L3) (inf3.1.3.1) 6. Discuss the benefits of using the problem-solving process. 7. A key benefit of using the process. (L1) (inf3.1.1.3) 8. An outlien of the benefits of using the process. (L2) (inf3.1.2.2) 9. A brief explanation of the benefits. (L3) (inf3.1.3.2) 10. A complete discussion of the benefits. (L4) (inf3.1.4.1) 11. Use the problem-solving process to complete the task below; 12. Identify a problem that requires a computer program solution. (L1) (inf3.1.1.4) 13. Decompose the problem into solvable components. (L2) (inf3.1.2.3) 14. State a single statement in plain English to present the solution for a major part of the problem. (L1) (inf3.1.1.5) 15. Describe the solution to the major part of the problem in plain English. (L2) (inf3.1.2.4) |
| Assignment | IA 2 :   |  |  |  |  | | --- | --- | --- | --- | | **SLO** | Specific Learning Outcomes (SLO): *Students are able to* | SKILL LEVEL | **SLO CODE** | | 17. | Use a flowchart to present the flow of the solution to a problem. | 2 | inf3.2.2.4 | | 12. | Use modular coding practices correctly. | 2 | inf3.3.2.2 | | 16. | Write detailed comments in the code to clarify the purpose of a piece of code. | 2 | inf3.3.2.4 | | 22. | Produce required program solution with all required coding elements using a programming tool. | 4 | inf3.3.4.1 | |
| Assessment | |  | | --- | | **Title: Programming Project**  **Starting Date:** June 3, 2020 **Due Date:** June 27, 2020  **Total Hours Spent:** 15 Hours (3 weeks)   * 5 hours of planning and draft coding (do a flow chart to guide the program) * 5 hours for practice coding and testing * 5 hours for coding and testing   **IA Location:** School Computer Lab | | You are to:   1. Identify a problem that requires a program solution [with the help of the teacher]; 2. Decompose the problem into main components; 3. Express the solution to the problems using pseudocode; 4. Express the logic of the program using a flowchart; 5. Code the program using current practices    * Modular    * Declaring of variables    * Using Control structures    * Comments 6. Test the code to show that it is working accurately; 7. Document all the stages of the task. 8. Save the program and a report showing all the work carried out in this project;   Hand into the teacher the saved copy of the program and the report | |
| Reference ClipartReferences | http://www.php.net/tut.php  http://www.php.net/  http://winlamp.sourceforge.net/ |



**WEEKLY CHECKLIST For Parents**:

Term: 2 Week number 4-8 Date: 8 June to 3 July Month: July

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Subject** | **Number of lessons** | **Days** | **Tick when activity is**  **complete** | **Parents comment** | **Signature** |
| **ICT Programming** | **1-5**  **Wk 4** | **Monday - Friday** |  |  |  |
| **ICT**  **Programming** | **1-5**  **Wk5** | **Monday - Friday** |  |  |  |
| **ICT**  **Programming** | **1-5**  **Wk 6** | **Monday - Friday** |  |  |  |
| **ICT**  **Programming** | **1-5**  **Wk7** | **Monday - Friday** |  |  |  |
| **ICT**  **Programming** | **1-5**  **Wk8** | **Monday - Friday** |  |  |  |

Lesson Plan

|  |  |
| --- | --- |
| G:\Home Learning Packages\Documents for SHEFA Schools Principal\teacher-computer-icons-school-test-education-teaching.jpg Teacher | Name : Rensy Ilaisa  Subject : ICT |
| G:\Home Learning Packages\Documents for SHEFA Schools Principal\download.jpg  Date | July 13 – August 4, 2020 |
| G:\Home Learning Packages\Documents for SHEFA Schools Principal\title.jpg | Topic : Microprocessor Programming |
| Learning outcomesLearning outcomes | * Define microprocessor*.* * Define machine code*.* * Outline the features of a microprocessor. * Describe the operation of a microprocessor in terms of input, processing, storage, and output. * Describe the interaction between processing and storage in the microprocessor. * Describe the process involved in converting high-level language to machine language. * Discuss the process involved in converting high-level language to machine language. |
| TopicIntroduction | **1. Microprocessors**  Definition: A microprocessor or a processor for short is a tiny electronic chip found inside a computer's central processing unit (CPU) and other digital devices.  Definition: Machine language or machine code is the lowest-level programming language. Machine languages are the only languages understood by computers. Consists of string of binary digits 0 & 1 and can process one low level task at a time; hardware dependent; no translations required. |
| Learners notes 1  Learners notes | **Basic features of a microprocessor**  **Cost:**The most important characteristics of a microprocessor is its low-cost availability. Because of the widespread use of microprocessors, the volume of production is very high. That’s why microprocessors chips are available at fairly low prices.  **Size:**The second key feature of the microprocessor is its small size. As the improvement in fabrication technology, its size does not even exceed a few inches on any side, even in the packaged form.  **Power consumption:**Another important feature is its low power consumption. They are manufactured by Metal-oxide semiconductors which require low power to operate.  **Versatility:**The versatility of a microprocessor results from its stored program. With the same hardware, it can be configured with many applications which also makes it flexible.  **Reliability:** Another advantage of this device is its extreme reliability. It has been found that the failure rate of an IC is relatively low at the packaging level.  **Operationof a microprocessor**  The basic operation of a microprocessor involves the following steps;  Image result for basic operation of a processor**Fetch:** Obtain INPUT instructions or data from memory  **Decode:** Translate instructions into commands  **Execute:** carryout command  **Store:** write result into memory   * Controlling all other parts of the machine and sending timing signals * Fetching data and instructions from memory. * Decoding instruction. * Executing programs stored in memory. * Performing communication among the Input/output (I/O) devices etc. * Based on instruction the processor fetches, if required, retrieving data from main memory or I/O module. * The instruction is then executed which may require performing arithmetic or logical operations on data. * Supervises and controls I/O devices. * Transfer of data to memory or an I/O Module.   **Processing and storage**  The instructions and data being processed by a processor or the CPU interact closely with the primary storage or the main memory. Memory stores program instructions or data for only as long as the program is in operation.  Most types of memory only store items while the device is turned on; data is destroyed when the device is turned off.If more than one program is running at once (often the case on large computers and sometimes on small computers), a single program can’t lay exclusive claim to memory.There may not be room in memory to hold the processed data.  The control unit of the processor fetch the instructions and data from input into the memory. And when required sends these items from memory to the arithmetic/logic unit, where an arithmetic operation or logical operation is performed. After being processed, the information is sent to memory, where it is hold until it is ready to be released to an output unit.  The main characteristic of memory is that it allows very fast access to instructions and data, no matter where the items are within it.  **2. High and Machine language**  Machine language also called machine code is a computer language that is directly understandable by a computer's CPU (central processing unit), and it is the language into which all programs must be converted before they can be run.  Machine code is extremely difficult for humans to read because it consists merely of patterns of bits (i.e., zeros and ones).  The general purpose of a High-Level Language (HLL) is to provide an easy to understand programming platform for developing computer programs. Such languages are considered high-level because they are closer to human languages and easier to understand. The first high-level languages were introduced in the 1950's. Today, there are many high-level languages in use, including BASIC, C, C++, Cobol, FORTRAN, Java, Pascal, Perl, PHP, Python, Ruby, and Visual Basic.  The general purpose of Low-level languages is to directly operate and handle the entire hardware of a digital device. Low Level languages can manage and manipulate computing hardware and components without any interpretation or translation. Programs written in a low-level language are directly executed on the hardware.  Examples of low-level language are Machine Language and Assembly Language. Machine language uses binary coding while Assembly Language uses mnemonic coding which is a symbolic representation of the machine code.  Image result for example of machine and assembly language  Machine language, or machine code, is the only language that can be understood by the computer directly. The instructions are written in binary as string of 1s and 0s as shown on the diagram above.  To write instructions in binary notation is a very difficult and will be a long process for humans. So, an assembly language was developed with some defined symbols and letters that can be easier for humans to understand. The assembly code requires a translator program also known as an assembler to convert the symbols into binary notation.  Many of the earliest computer programs were written in assembly languages. Most programmers today don't use assembly languages very often, but they are still used for applications like operating systems of electronic devices and technical applications. While easier than machine code, assembly languages are still pretty difficult to understand. This is why high-level languages have been developed.  A high-level program is translated into machine code using a **compiler or interpreter**. The compiler is used to translate source code into machine code or compiled code. This does not yet use any of the input data. When the compiled code is executed, referred to as 'running the program,' the program processes the input data to produce the desired output. Instructions written in a high-level language are translated to the machine level language  using the binary digits “0” and “1”. The codes are then executed and converted to electrical signals in the form “ON” and “OFF” to instruct the device to do what the program wants.  **3. Microprocessor controls**  A microprocessor is used to control computers and many digital devices. A **programmable microprocessor**or known as a **microcontroller**can be programmed to perform different functions making it possible for the development of many different electronic devices but using the same programmable microprocessors. Such possibility allows for cheaper and efficient productions of many different types of electronic components and devices.  A microcontroller used by a digital device is programmed using a high-level language to perform specific tasks. The high-level code is then translated into the machine code using a compiler and then saved into the microprocessor and embedded into a digital device. The program provides the instructions for the microprocessor or microcontroller to control the function of the device.  Examples of devices being controlled by microprocessors;   * Computer system; the CPU controls all the processing that go's on within the computer. * Washing machines; microprocessors control things like water temperature, valves to let water in, valves to let water out, etc. * Alarm systems; microprocessor is used to detect intruders and set the alarm off in the burglar alarm. * Heating systems; microprocessors are used to control things e.g. when the heating is switched on/offkeeping a constant temperaturesome can make the temperature different from room to room * Intelligent ovens and microwaves; the microprocessor inside can read the barcodes on the package and then automatically set the temperature and cooking time to the perfect level. * Intelligent fridges; keep food at optimum temperatures and help reduce waste.   The advantage of the microprocessor is that one device can control everything. The disadvantage comes if the microprocessor malfunctions: all the systems could be affected, and the replacement cost is high. With microcontrollers, on the other hand, if one is damaged it can be easily replaced and when one fails the others continue to function. |
|  | <https://www.youtube.com/watch?v=LnzuMJLZRdU>  <https://www.youtube.com/watch?v=htVJTW50WA0&list=PLWdpu3kSIiK0toH7d3XrSgP9Y69nPFcKI> |
|  | 1. Define microprocessor. (L1) (inf4.1.1.1) 2. Define machine code. (L1) (inf4.1.1.2) 3. Outline the features of a microprocessor. 4. A given feature of a microprocessor. (L1) (inf4.1.1.3) 5. An outline of the features. (L2) (inf4.1.2.1) 6. Describe the operation of a microprocessor. 7. A component of the operation of a microprocessor provided. (L1) (inf4.1.1.4) 8. A full description of the operation. (L2) (inf4.1.2.2) 9. Explain the interaction between processing and storage in the microprocessor. 10. A part of the interaction is provided. (L1) (inf4.1.1.5) 11. A full description of the interaction. (L2) (inf4.1.2.3) 12. A complete explanation of the interaction. (L3) (inf4.1.3.1) 13. Discuss the process involved in converting high-level language to machine language. 14. A part of the conversion process provided. (L1) (inf4.1.1.6) 15. The conversion process described fully. (L2) (inf4.1.2.4) 16. A brief explanation of the conversion process. (L3) (inf4.1.3.2) 17. A complete discussion of the conversion process. (L4) (inf4.1.4.1) 18. Discuss the process involved when a microprocessor controls the hardware. 19. A part of the controlling process given. (L1) (inf4.1.1.7) 20. An outline of the controlling process. (L2) (inf4.1.2.5) 21. An explanation of the controlling process. (L3) (inf4.1.3.3) 22. A complete discussion of the controlling process. (L4) (inf4.1.4.2)   . |
| Assignment | |  |  |  |  | | --- | --- | --- | --- | | **SLO** | Specific Learning Outcomes (SLO): *Students are able to* | SKILL LEVEL | **SLO CODE** | | 28. | Identify an appropriate example of a critical change in the physical environment of a specific embedded device. | 1 | inf4.2.1.13 | | 11. | Design a flow chart to present the logical solution to the identified task. | 2 | inf4.2.2.3 | | 22. | Produce a high-level program instructing a programmable microprocessor to respond when a critical change in the physical environment is detected. | 3 | inf4.2.3.4 | | 15. | Produce a high-level program for a programmable microprocessor to perform a specific task. | 4 | inf4.2.4.1 | |
| Assessment | .   |  | | --- | | **Title: Microprocessor Control Project**  **Starting Date:** July 13, 2020  **Due Date:** August 3, 2020  **Total Hours Spent:** 15 Hours (3 weeks)   * 5 hours for planning and discussion * 5 hours for testing the relationship between programming and machine language * 5 hours write up   **IA Location:** School Computer Lab/PC/Laptop at home | | You are to:   1. Use a microprocessor kit to meet the requirement of the task; 2. Identify an appropriate task that requires a programmable microprocessor to perform; 3. Design a flowchart to present the logic of the task to be performed; 4. Write a high-level code to manipulate a microprocessor to perform a simple task; 5. Identify an example of physical change that can affect an embedded device to respond 6. Write a high-level code to manipulate a microprocessor to respond to a change in the physical environment of the embedded device; 7. Capture the work on a video for confirmation of completed task; 8. Hand into the teacher the saved copy of the program and proof of a working microprocessor. | |
| Reference ClipartReferences | 1. Phishing resource material is at http://en.wikipedia.org/wiki/Phishing 2. Encryption resource material can be found at   http://www.commoncriteriaportal.org/files/epfiles/st\_vid3008-st.pdf |

Term: 2 Week number: 9-13 Dates July 13 to August 4 Month: August

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Subject** | **Number of lessons** | **Days** | **Tickwhenactivityiscomplete** | **Parents comment** | **Signature** |
| **ICT – Microprocessor Programming** | **1-5**  **Wk9** | **Monday-Friday** |  |  |  |
| **ICT - Microprocessor Programming** | **1-5**  **Wk10** | **Monday-Friday** |  |  |  |
| **ICT - Microprocessor Programming** | **1-5**  **Wk11** | **Monday-Friday** |  |  |  |
| **ICT - Microprocessor Programming** | **1-5**  **Wk12** | **Monday-Friday** |  |  |  |
| **ICT - Microprocessor Programming** | **1-5**  **Wk13** | **Monday-Friday** |  |  |  |