Information technology task

Technology

What are the learners required to do and provide? 35 How will learners go about it? 35 Skills required 35 What must the learners be taught beforehand? 36 Malpractices Learner authentication of the PAT Role of the teacher 37 Supervised/Controlled conditions Managing the PAT 38 Assessment Evidence 38 Requirements 38 Non-compliance 39 What the PAT? 6 37 The PAT is a software development project in which you will have an opportunity to demonstrate your programming skills and understanding of the relationship between the different areas of solution development. You will also be required to monster your knowledge and understanding of the software development life cycle through analysis, design, coding and testing of your project. You will have to show effective use of the software design tools and techniques which you have studied.

You need to provide the following outputs: A report where you (Phase 1) discuss the research/investigation done regarding the project provide a brief description of the purpose and scope of your project provide analysis of a possible solution a document outlining the system design (Phase 2) a working Scratch program, fully documented, that implements the planned solution Phase 3) Note: You will also be required to demonstrate and discuss your program during a debriefing session.

Mark Allocation Development Phase Max Mark Phase 1 Analysis 27 Phase 2 Design Phase 3 Coding and Testing Complexity Level General Final product and impression 16 Total 155 The PAT counts 25% of your final mark for IT, therefore it is crucial that you strive to produce work of a high standard. The PAT is a compulsory component of the final end-of-year examination for IT.

You need to complete your PAT before you start the Grade 10 end-of-year examination.

Not submitting your PAT or any part of the PAT, ill mean that you will be awarded a zero (" O") for the PAT component of the examination or the parts not submitted. Topic Educational Software Educational software are programs that are designed to help people to learn about certain subjects, expand concepts, reinforce development, understand a concept, or assist them in learning specific subject content or a skill as they use the program.

Considering the requirements specified in this document (see PAT Requirements, up), choose any school subject and develop a Delphi/Java educational software program to support teaching and learning in South African schools. Ideas for educational software: An educational game Drilling mathematical concepts such as addition, subtraction, multiplication and division in lower grades Drilling vocabulary/correct spelling in languages Science simulation/experiment Subject terminology interactive quiz Unfolding shapes such as cones, prisms, etc. Lactating circumference, area and volume Teaching kids to read time, calculate time, work with money, etc. Etc. Your final program must comprise one single, logically related piece of software. Projects that consist of two or more unrelated programs will only obtain marks for en of the parts since only one of the programs will be regarded as the actual project. Note: The quality of the programming code that manipulates data successfully according to user requirements will greatly influence your project mark. Quantity will not replace variety, effectiveness and quality.

Overview Phase 1 - analysis The purpose of Phase 1 is to determine what should be done and what the user requirements are: Research/investigate the topic/scenario/the potential user(s) to gather facts about the nature of the program you intend to produce Define the task Determine the requirements Formulate acceptance tests hash 2 - Design The purpose of Phase 2 is to determine how the program/system will meet the requirements and to plan and design the solution to the problem: Clarify the requirements, indicating how your solution/program will meet each requirement/ goal Design the solution, clearly indicating the logical program flow and navigation between screens/scenes Design the GIG-II(s) Define the input, processing and output Design validation and test strategies Define variables and lists and their use phase 3 - coding and testing The purpose of Phase 3 is to implement the design by writing the code and testing he program: Write the programming code to implement the design and to complete the program.

Test and debug the program Clarify sections of the code by adding comments Write project notes for the program Demonstrate your program and answer questions about the program and the code during a debriefing session PAT Requirements The project should include the following, appropriately integrated: A multi-screen/ scene Graphical User Interface (GU') Variables and lists Manipulation/transformation of data through Mathematical/statistical processes String/text processes GUI The GUI must be functional and based on sound human computer interaction (HCI) principles. The GUI must have at least an opening screen (start-up screen, e. G. Menu), closing screen (e. G. Outcome) and three other screens (five in total). Variables and Lists Use appropriate, well-named variables and lists Carefully consider the scope of variables (this sprite vs.. All sprites) Modular Programming More When I receive blocks than Broadcast blocks speaks to reuse of code. Further requirements Apply good programming principles and techniques: Descriptive names for sprites, variables, lists, etc.

Well-structured, readable code Add comments to explain pieces of code, especially on the manner in which input variables/lists and output variables/lists are used. Write project notes: Describe what the program does Describe how to use/interact with the program General programming aspects that will be assessed: Programming style Graphical user interface (GU') Use of human-computer interaction (HCI) and software engineering principles Functionality of the program Robustness of the program, including effective algorithms and the use of defensive programming techniques Level of expert programming Whether the project matches the original aims and goals

Internal documentation to explain sections of the program To complete the tasks, you will need Scratch programming software Word processing software Internet access to find data and information Access to other sources such as printed media (e. G. Magazines, newspapers, brochures, textbooks) or other electronic material such (e. G. E-books, e-articles) Access to facilities to convert hard copies to electronic documents, e. G. Scanner, digital camera Storage media to store and backup your work electronically, e. G. Flash drive, rewritable CD/DVD or access to cloud storage (Noontide, Drop, etc.) Misconduct As the PAT is an individual project that is part of your final promotion mark, you may not: get help from others without acknowledging this help submit work which is not your own, e. G. Regaining code that was written by someone else lend your PAT work to other learners allow other learners to access or use your own work (this does not mean that you may not lend or borrow books to or from another learner, but you may not plagiarism other learners' research) include work directly copied from books, the internet or other sources without acknowledgement and recognition (this may not exceed 20% of the ark you submit) The above actions constitute misconduct, for which a penalty will be applied. Non-compliance You will be given the opportunity to submit outstanding work or present yourself for the PAT as outlined. Should you fail to fulfill any Practical Assessment Task (PAT) requirements, you will be awarded a zero (" O") for the the outstanding part or for the entire PAT (should all parts be outstanding).

Glossary - Terms and Examples See Manicure F for Glossary of Terms See Manicure G for an example Instructions for Phase 1 The aim of Phase 1 is to get a clear understanding of the problem/task at hand fine the task in your own words determine what the programming solution should do and provide (what functionalities/features must be built into the program) determine when a user will know that a functionality/feature is successfully implemented The deliverable of this phase is a report based on your investigation/research (See Manicure A) a document (see Manicure B) that clearly explains what the problem/task is and what the solution should be capable of doing in terms of the needs of the user(s), the stated objectives and how the user(s) will confirm that the needs have been met. Do Research The aim is to gather facts about the topic and the nature of the program you are developing.

Your investigation/research should help you to understand the topic/scenario clarify the nature/type of program you are required to develop and the potential user(s) look at some existing solutions, where possible, to gather ideas look for similarities, differences missing features general working/flow/interaction understand what particular type of program(s) is suitable for this project For this project (educational software) it means that you will have to Investigate existing educational software and how they work The outcome of the research is a report (B± 800 words) that discusses, for example, features of existing solutions including similarities, differences, missing features and the general working/flow/interaction what a software solution for the problem may include Use clear, unambiguous language. Define the task The aim is to provide an overall picture of the purpose and scope of the project but not the details. Write a brief description (B±1 50 words) in your own words to describe, in general terms, the problem/task and indicate how the project will solve the problem. In other words, the description should convince the user that you understand the needs/gaps/problems that your solution will address the needs/gaps/problems The user must look forward to using your program. Your description must also address the PAT requirements/specification.

Do the The aim is to determine who will use the program determine the user requirements for the program determine when one would know that a required functionality has been successfully implemented It should specify WHAT is needed (not HOW). Specify and document (See Manicure B): The prospective user(s) - Who will use the system? The user is the target audience (people that will use the program) that will determine the needs or the goals of the system (in the case of the PAT, you may put yourself in the user's shoes or ask a friend what they would want from such a program). The user stories - Tells the designer/programmer what the user wants. What will the users do with the system? /What are the users' needs? That goals does a user want to achieve? ' What should the programming solution do and provide? Indicate the functionalities/features that must be built into the system.

Example: As a learner I want to practice my subtraction skills so that I can improve my Math ark WHO As WHAT I want to WHY So that User/Actor/role Goal/program feature required Value or benefit User stories tell the programmer what the user wants and therefore define the user requirements must be independent, I. E. Each user story can be developed, tested on their own and do not depend on others Note: Each user story/goal could possibly represent a scene. From the user stories, identify the goals. A goal represents a functionality (functional requirement) that can be used or performed in isolation (is independent), I. E. A user will be able to request only that service/function in a single session. State these goals in the format, e. G.

Practice subtraction skills (named by verb) Use a diagram to provide a graphical representation of the goals/functionalities/features of the system (at top level) - all that the program will be able to do/that can be done with the system (in a single session). As a guideline, there should be a sufficient number of goals to create three screens (at least three goals). Formulate acceptance tests - when will the user know that a goal is achieved/a functionality is successfully implemented? Acceptance tests are derived from user stories/goals. Each user story/ AOL should have at least one acceptance test. Example: I know this is successful/achieved when I see the outcome of my subtraction practice actor Verb/ action Observable result Hand In Your teacher will provide the date on which you must hand in Phase 1 of the PAT.

Once you have completed Phase 1 of the project, submit the following: A report Outlining the research/investigation (B±800 words) A document (See Manicure B) with the task description (B±1 50 words) specifying the prospective user(s) providing the user stories (requirements/goals) providing a graphical representation of the goals utilizing the user acceptance tests Your declaration for Phase 1 (Manicure D) Instructions for Phase 2 determine HOW you will go about solving the problem and plan the detail. Produce a plan showing a high level overview of how the solution will be constructed. Use explained/annotated pseudopodia/diagrams (or suitable alternative). The plan must show all the main blocks within the proposed solution. Pacify and document an overall design that meets the requirements, using software design tools such as PIP- diagrams, flow charts, GUI sketches/prototypes, clearly annotated. Design the Solution GUI/Scenes, Input, Output, Processing (Algorithms), Variables/Lists, flow, etc.) The aim is to clarify/refine the user stories/goals and fill in detail design the GUI (screens) determine a test strategy to prevent input-processing-output (PIP) errors Use appropriate design tools and techniques to design the overall solution considering all constituent parts and the inter-relationships between the various parts of the program. Clarify/Refine User Stories (Requirements) The aim is to provide substance to the user stories/goals.

This is generally achieved by participative design (conversations between user and signer/programmer) and captured as additional notes that provide clarity. Break down each user goal (from Phase 1) into sequences of executable steps/actions or events necessary to achieve the goal. Firstly, describe the essential steps/actions/ events to achieve the goal. The essential steps represent the shortest path or flow of events to success/to achieve a specific goal (from the moment the actor initiates/ triggers it until the goal is achieved while everything goes correctly). Example of essential steps for Practice subtraction skills goal: Learner (user) Program 1. Displays subtraction problem 2. Enter answer 3. Check answer 4.

Display message 5. Determine/Calculate outcome of exercise Secondly, determine the additional (alternative) steps/actions that will be performed if something goes wrong, e. G. If incorrect answer is submitted, etc. Example of additional steps (What could cause goal not to be achieved?): Incorrect answer additional steps: Learner AAA Incorrect answer Display message/produce sound Provide another chance to enter answer Count number of attempts to correct answer The additional steps will help you to determine tests that you/the program have to perform to ensure a robust program. Note: The sequence of steps for each goal could be presented in a flow chart, e. G.

The aim is to design a GUI (screens) that considers good human-computer-interface (HCI) principles: The user - type and context Appropriate, effective input and output strategies to address requirements/needs Dialogue - must be relevant, simple and clear Sprite/object usage and presentation - well selected and relevant, well placed and clear purpose Helpful error messages/feedback - neat, correctly formatted, clear and well-presented Exits - clearly marked, placed appropriately Sensible usage of space on the screen/stage reverts input errors minimizes the amount of information a user has to input Plan and design each scene according to the user stories/goals identified in Phase 1 . For each screen/stage, use the Phase 2 template (Manicure B) and indicate the following (where applicable): Name of the screen/scene A sketch of the screen/scene Navigation (previous screen/scene, next screen/scene, branching to another screen/ scene) Background Sprite(s) What the user will see (e. G. Read), hear and do (click, type, etc. For each object (background/sprite) used as part of the screen/scene, indicate the following (where applicable) associated with the object: Costume(s) Responsibilities and function Collaborators Input and Output What it will broadcast What it will receive Variables/lists: the name, type of data Algorithms (processing) Provide example(s) of planned data capture/data entry (input) designs and of planned, valid output designs (prototype screen dumps may be used but must be annotated). Devise Tests The aim is to develop and document a test strategy to ensure data integrity, I. E. Defensive programming techniques to avoid input, output and processing errors: What must be tested? Why must it be tested? When will it be tested? How will it be tested? Use the steps (generally the additional steps/actions/events) to derive test cases. Test cases must be executable. Provide suitable test inputs, e. G. Test data (normal (typical) data, erroneous data and boundary data) Provide expected results for normal (typical) data, erroneous data and boundary data.