Appendices
APPENDIX A
SIGNING UP AND CONSENT
Dear user,

During the study, you will be asked to do the following activities:

1. Register yourself as a user in the website.
2. Do a pre-test
3. Perform two “learning styles” tests.
4. Visit and explore a short lessons.
5. Fill out a post-test (Questionnaire).

End the experiment

This experiment is for both: None-Computer and Computer fields. Both of them can do it.

We wish to ensure that you understand your rights as a subject and that your consent to participate in this survey is fully informed. Your rights are listed below. Once you review this list, if you still wish to participate, please click the SURVEY button. By completing the registration, you consent to participate.

1. You have the right to terminate your participation in this research at anytime.
2. You have the right to know that the responses you offer will be used in a PhD research.
3. The information obtained during this survey will remain confidential, to the extent allowed by law.

Please consent to participation by completing the registration. If you wish to withdraw, please close this page.
Adaptive system to learning styles

SIGN UP FOR REGISTRATION

All fields marked with (*) are compulsory

- Username:
- Password:
- Confirm Password:
- Email:
- Country:
- State:
- City:
- Gender: Male Female
- Discipline: Computer field None Computer field
- Field:
- University:
- Experience in software development: 1-2 years 3 years 4+ years
- Age:
- Status:
- I agree to participate in the study

Submit

Dear user,

During the study, you will be asked to do the following activities:
1. Register yourself as a user in the website.
2. Do a pre-test.
3. Perform two "learning styles" tests.
4. Visit and explore a short lecture.
5. Fill out a post-test (Questionnaire).

End the experiment.

This experiment is for both : None-Computer and Computer fields. Both of them can do it.

We wish to ensure that you understand your rights as a subject and that your consent to participate in this survey is fully informed. Your rights are listed below. Once you review this list, if you still wish to participate, please click the SIGNUP button. By completing the registration, you consent to participate.

1. You have the right to terminate your participation in this research at anytime.
2. You have the right to know that the responses you offer will be used in a PHP research.
3. The information obtained during this survey will remain confidential, to the extent allowed by law.
4. Please consent to participation by completing the registration. If you wish to withdraw, please close this page.

This work is based on an ongoing research project
School of Mathematical and Computer Sciences,

255

Appendices
Adaptive system to learning styles

Educational studies have shown that discovering and applying Learning Style can help to improve academic performance.

On entering our site you will be asked to complete two tests to determine your most effective Learning Style. This will enable you to discover your learning styles.

This work is based on an ongoing research project
School of Mathematical and Computer Sciences,
UK
APPENDIX B
LEARNING STYLES AND STUDENT’S PROFILE
VARK

ADAPTIVE SYSTEM TO LEARNING STYLES

Welcome, bbbbb1

Logout, Home, Help

Vark Test

Q.1  YOU ARE ABOUT TO GIVE DIRECTIONS TO A PERSON WHO IS STANDING WITH YOU. SHE IS STAYING IN A HOTEL IN TOWN AND WANTS TO VISIT YOUR HOUSE LATER. SHE HAS A RENTAL CAR. I WOULD:

a)  ☐ draw or provide a map on paper.

b)  ☐ tell her the directions.

c)  ☐ write down the directions (without a map).

d)  ☐ collect her from the hotel in a car.

Q.2  YOU ARE NOT SURE WHETHER A WORD SHOULD BE SPELLED (DEPENDENT) OR (DEPendant). I WOULD:

a)  ☐ look it up in the dictionary.

b)  ☐ see the word in my mind and choose by the way it looks

c)  ☐ sound it out in my mind.

d)  ☐ write both versions down on paper and choose one.

Q.3  YOU HAVE JUST RECEIVED A COPY OF YOUR ITINERARY FOR A WORLD TRIP. THIS IS OF INTEREST TO SOME FRIENDS. I WOULD:

a)  ☐ phone, text or email them and tell them about it.
b) ☐ send them a copy of the printed itinerary.

c) ☐ show them on a map of the world.

d) ☐ describe what I plan to do at each place on the itinerary

Q.4 You are going to cook something as a special treat for your family. I would:

a) ☐ cook something familiar without the need for instructions.

b) ☐ thumb through the cookbook looking for ideas from the pictures.

c) ☐ refer to a specific cookbook where there is a good recipe.

Q.5 A group of tourists has been assigned to you to find out about wildlife reserves or parks. I would:

a) ☐ drive them to a wildlife reserve or park.

b) ☐ show them slides and photographs.

c) ☐ give them pamphlets or a book on wildlife reserves or parks.

d) ☐ give them a talk on wildlife reserves or parks.

Q.6 You are about to purchase a new CD player. Other than price, what would most influence your decision?

a) ☐ the salesperson telling you what you want to know

b) ☐ reading the details about it.

c) ☐ playing with the controls and listening to it.

d) ☐ it looks really smart and fashionable.
Q.7  **RECALL A TIME IN YOUR LIFE WHEN YOU LEARNED HOW TO DO SOMETHING LIKE PLAYING A NEW BOARD GAME. TRY TO AVOID CHOOSING A VERY PHYSICAL SKILL, E.G. RIDING A BIKE. I LEARNT BEST BY:**

a) visual clues -- pictures, diagrams and charts.

b) written instructions.

c) listening to somebody explaining it.

d) doing it or trying it.

---

Q.8  **YOU HAVE A KNEE PROBLEM. I WOULD PREFER THAT THE DOCTOR:**

a) told me what was wrong.

b) showed me a diagram of what was wrong.

c) used a model of a knee to show me what was wrong.

---

Q.9  **YOU ARE ABOUT TO LEARN TO USE A NEW PROGRAM ON A COMPUTER. I WOULD:**

a) sit down at the keyboard and experiment with the program.

b) read the manual that came with the program.

c) telephone or text a friend and ask questions about the program.

---

Q.10  **A NEW MOVIE HAS ARRIVED IN TOWN. WHAT WOULD MOST INFLUENCE YOUR DECISION TO GO (OR NOT GO)?**

a) I heard a review about it on the radio.

b) I read a review about it.

c) I saw a preview of it.
Adaptive system to learning styles

Welcome, bbbbb1

Logout, Home, Help

Honey and Mumford Test

Q.1 I tend to solve problems using a step-by-step approach.

a) True

Appendices
Q.2 I HAVE A REPUTATION FOR SAYING WHAT I THINK, SIMPLY AND DIRECTLY.

a) True

b) False

Q.3 I LIKE THE SORT OF WORK WHERE I HAVE TIME FOR THOROUGH PREPARATION AND IMPLEMENTATION.

a) True

b) False

Q.4 WHAT MATTERS MOST IS WHETHER SOMETHING WORKS IN PRACTICE.

a) True

b) False

Q.5 I DON’T LIKE DISORGANISED THINGS AND I PREFER TO FIT THINGS INTO A COHERENT PATTERN.

a) True

b) False

Q.6 I ACCEPT AND STICK TO LAID DOWN PROCEDURES AND POLICIES SO LONG AS I REGARD THEM AS AN EFFICIENT WAY OF GETTING THE JOB DONE.

a) True

b) False
Q.7  I LIKE TO RELATE MY ACTIONS TO A GENERAL PRINCIPLE.
   a)  □ True
   b)  □ False

Q.8  IN DISCUSSIONS I LIKE TO GET STRAIGHT TO THE POINT.
   a)  □ True
   b)  □ False

Q.9  I TEND TO HAVE DISTANT, RATHER FORMAL RELATIONSHIPS WITH PEOPLE AT WORK.
   a)  □ True
   b)  □ False

Q.10 I ENJOY FUN-LOVING, SPONTANEOUS PEOPLE.
     a)  □ True
     b)  □ False

Q.11 I TEND TO BE OPEN ABOUT HOW I AM FEELING.
     a)  □ True
     b)  □ False

Q.12 IN DISCUSSION I ENJOY WATCHING THE MANOEUVRINGS OF THE OTHER PARTICIPANTS.
     a)  □ True
Q.13 I PREFER TO RESPOND TO EVENTS ON A SPONTANEOUS, FLEXIBLE BASIS RATHER THAN PLAN THINGS OUT IN ADVANCE.

a) False

Q.14 IT WORRIES ME IF I HAVE TO RUSH OUT A PIECE OF WORK TO MEET A TIGHT DEADLINE.

a) True

Q.15 QUIET, THOUGHTFUL PEOPLE TEND TO MAKE ME FEEL UNEASY.

a) True

Q.16 I OFTEN GET IRRITATED BY PEOPLE WHO WANT TO RUSH THINGS.

a) True

Q.17 IT IS MORE IMPORTANT TO ENJOY THE PRESENT MOMENT THAN TO THINK ABOUT THE PAST OR FUTURE.

a) True

b) False
Q.18  **In discussions I usually produce lots of spontaneous ideas.**

a) ☐ True

b) ☐ False

Q.19  **I prefer to stand back from a situation and consider all the perspectives.**

a) ☐ True

b) ☐ False

Q.20  **I tend to discuss specific things with people rather than engaging in social discussion.**

a) ☐ True

b) ☐ False

Q.21  **If I have a report to write I tend to produce lots of drafts before settling on the final version.**

a) ☐ True

b) ☐ False

Q.22  **I am keen to reach answers via a logical approach.**

a) ☐ True

b) ☐ False

Q.23  **I enjoy being the one that talks a lot.**
Q.24  IN DISCUSSIONS I OFTEN FIND I AM THE REALIST, KEEPING PEOPLE TO THE POINT AND AVOIDING WILD SPECULATIONS.

a) ☐ True
b) ☐ False

Q.25  IN DISCUSSIONS WITH PEOPLE I OFTEN FIND I AM THE MOST DISPASSIONATE AND OBJECTIVE.

a) ☐ True
b) ☐ False

Q.26  I LIKE TO BE ABLE TO RELATE CURRENT ACTIONS TO A LONGER TERM BIGGER PICTURE.

a) ☐ True
b) ☐ False

Q.27  WHEN THINGS GO WRONG I AM HAPPY TO SHRUG IT OFF AND ‘PUT IT DOWN TO EXPERIENCE’.

a) ☐ True
b) ☐ False

Q.28  I TEND TO REJECT WILD, SPONTANEOUS IDEAS AS BEING IMPractical.

a) ☐ True
b) False

Q.29 IT’S BEST TO THINK CAREFULLY BEFORE TAKING ACTION.
   a) True
   b) False

Q.30 ON BALANCE I DO THE LISTENING RATHER THAN THE TALKING.
   a) True
   b) False

Q.31 I TEND TO BE TOUGH ON PEOPLE WHO FIND IT DIFFICULT TO ADOPT A LOGICAL APPROACH.
   a) True
   b) False

Q.32 MOST TIMES, I BELIEVE THE END JUSTIFIES THE MEANS.
   a) True
   b) False

Q.33 I DON’T MIND HURTING PEOPLE’S FEELINGS AS LONG AS THE JOB GETS DONE.
   a) True
   b) False

Q.34 I DO WHATEVER IS EXPEDIENT TO GET THE JOB DONE.
Q.35  I quickly get bored with methodical, detailed work.
   a)  ☐ True
   b)  ☐ False

Q.36  I am always interested to find out what people think.
   a)  ☐ True
   b)  ☐ False

Q.37  I like meetings to be run on methodical lines, sticking to a laid down agenda, etc.
   a)  ☐ True
   b)  ☐ False

Q.38  I steer clear of subjective or ambiguous topics.
   a)  ☐ True
   b)  ☐ False

Q.39  I enjoy the drama and excitement of a crisis situation.
   a)  ☐ True
   b)  ☐ False
Q.40  **People often find me insensitive to their feelings.**

a) ☐ True

b) ☐ False
APPENDIX C
TEACHING MATERIALS
Teaching Materials

Waterfall Model *(bullet points)*

Overview
In this lesson the student will:

- Understand the Waterfall lifestyle model of software development.
- Understand the concept of the waterfall lifecycle model.
- Identify the phases of the waterfall lifecycle model.
- Identify advantages and disadvantages of the Waterfall lifecycle model.

Introduction

- The waterfall model was developed by Winston Royce in 1970.
- The waterfall method is the most commonly used model of software development today.
- Waterfall model has five sequential stages or phases:
  - Requirements
  - Design
  - Implementation
  - Testing
  - Maintenance
- Progress is documented through each phase.
- Phases are closed/locked upon completion.
- Stages do not overlap.
- If changes are needed after a stage is completed it happens after completion of the cycle.

Explanation of the Waterfall Model

- Documentation is essential to the success of this model of software development.
  - Documents are created throughout each phase.
  - Software isn’t created until end phases.
  - Steps are linear and must remain in order.
  - Documents for each stage are closed/locked at the end of each stage before moving to the next stage.
- You can quickly find errors in early stages of development when familiar with program being developed. Examples:
  - Updating old software
  - Creating new version of an existing program that you are familiar with.
Stages or Phases of the Waterfall Model

- **Requirements**
  - Requirements are decided upon and documented.

- **Design**
  - Decisions on all levels of design are made.
  - Programs that are going to be used are decided upon.
  - What algorithms and data structures are going to be used.
  - Everything that is decided upon is documented.

- **Implementation**
  - Code is written in this phase.
  - Tests are run on small portions, most of the testing isn’t done in this phase.

- **Testing**
  - All aspects of the software are tested.
  - After primary testing to be sure the program works like it should customer is included in final testing.

- **Maintenance**
  - Improvements are made as needed.
  - Improvements go through same steps as taken with original software development.

Pros and Cons

**Pro**
- All steps are well defined.
- Documentation makes it possible for project managers to quickly tell if the project is on track or not.
- Works well when developing familiar software (i.e.: updating older software, creating a new program like something already on the market).
- Requirements are defined early in the process.
- Documentation makes it possible for developers to join the team late and still know what is going on.

**Con**
- Assumes initial information during requirements phase is complete.
- Assumes the flow from beginning to end is going to be one smooth, sequential motion which rarely happens.
- Implementation and testing do not happen until late in the process.
- Complications arise from not being able to make changes during the process including:
o Changes thought of after the requirements phase is closed are put off until the maintenance phase.
o Delays in completion often arise when requirements change or are incomplete.
- Client can get frustrated with the long wait period between the requirements phase and when they are able to see any progress, especially with larger projects.
- Early assumptions aren’t always accurate, especially if the software being developed is unfamiliar.
- Closed stages cannot be revisited.
- Requirements must be reviewed early and must be accurate.
- Incomplete user definitions in the beginning cannot be changed or added to later.
- It is nearly impossible to think of every requirement for new products.

Waterfall Model (paragraphs)

Overview
This lesson introduces students to the Waterfall life cycle model. On completion students will have an understanding of the waterfall model including concept, phases, advantages, disadvantages, and when to use the Waterfall life cycle model.

Introduction
Since the development of this Waterfall life cycle model in 1970 by Winston Royce, the waterfall life cycle model has become the most commonly used model of software development. In this model software development goes through five steps: requirements, design, implementation, testing, and maintenance.

The waterfall life cycle model has a unique way of showing progress in set stages or phases. Phases are set in a linear manner. Each step must be properly completed and documented before the next step can take place. Once a phase is completed with this model it is closed and not revisited. Instead of going back to previous steps, if changes are needed in a completed step they are saved until the end of the process.

Explanation of the Waterfall Model
During software production each stage is closed and documented. These documents are required before going to the next step. Steps are not
complete until they are properly documented. The next step cannot be started before finishing the previous step. Two phases are never worked on at the same time in the pure Waterfall model.

This model of software creation is favored when creating updated forms of existing software. With the waterfall model you can quickly find errors in early phases of this type of development. Questions that arise late on during development when working on new types of software can be costly.

Since software isn’t developed until the last stages the only thing tangible that is produced in the steps are documentation. When working on software updates, or new software that is to perform like existing software, the documents produced by each step show meaningful progress. When using this model to produce software unlike anything you have worked on before it is easy for some requirements to be overlooked or missed completely.

Not being able to return to previous phases to make any changes or corrections during development is a downfall of the waterfall model. The Salmon lifecycle model is another way of looking at the waterfall model allowing corrections to be done. Progress must go through each step in order so if you return to a previous step the work done after that point has to be redone increasing production time and cost. The flexibility it takes to make updates is time consuming and costly when using this method.

Copious amounts of documents and a lack of any other product causes some customers to become concerned about deadlines. When the project is on track there is little to show for the work that has been done besides paperwork until software development is near completion.

**Stages or Phases of the Waterfall Model**

This model begins with the requirements stage. As the name hints, requirements are decided upon and documented during this phase.

During the design phase decisions are made on all levels of design. This is where you decide how the program will work, what other programs it will work with and how, what algorithms are going to be used, data structures and so forth. Everything must be documented with requirements and explanations for those requirements so that all others working on the project, especially those coming in later in the process will understand.

The documentation quality during the design phase affects the difficulty level during the implementation phase. More detailed design documents leave little room for changes. Design documents with fewer details can slow you down during the implementation phase. If documentation is weak as to why a
certain choice was made in the design phase confusion and errors can happen further down the line during development.

The implementation or construction phase is where coding starts. Programmers using compilers write the program. During this phase the idea becomes more than just documents, the software is created. Small portions of the program may be able to be tested at this point, but for the most part that is left for the testing phase.

Once the implementation phase is finished the Testing phase begins. This is where debugging and profiling tools come in handy. During the testing phase all aspects of the software are tested. It is important the software performs as expected and up to the specifications of the customer. After primary testing to be sure the program works the way expected the customer gets involved in final testing to make sure the program is working as needed for its intended use.

Once the system is tested and found to meet the required specifications it is delivered and then the maintenance phase begins. With use clients usually find something that needs to work faster, better, or just differently than it is currently doing. Making improvements to the software once it is completed needs to go through the same steps taken with the original software. Since the programmer is now familiar with the software and how it works updates and improvements take less time than original development.

Pros and Cons of the Waterfall Model

PRO
Among the advantages of this method is the linear approach with well defined phases. Steps needed from beginning to the end of the development are all included and defined in this method. Extensive documentation makes it possible to define and measure progress through the process. Project managers can easily see where the progress stands by reviewing documentation. When you are developing an application to be like an existing application that you are familiar with this model works well.

CON
The waterfall model assumes that the flow of the project goes from beginning to end in one smooth, sequential motion. In reality this rarely happens.

Even though documentation is done throughout the early phases the lack of implementation and testing until the whole project is completed can cause costly delays due to changes that weren’t initially thought of during in the requirements and design phases. If changes are thought of before the project is finished the waterfall model doesn’t have in place a way to go back
to completed steps and make changes before completion of all of the steps is finished.

With large projects clients can become frustrated with the lack of tangible results. While clients are waiting for working versions of their software they are likely to find other requirements they hadn’t thought of earlier.

Assumptions made during the requirement and design phases aren’t always accurate or complete when developing unfamiliar software. Because the product isn’t actually tested until the end of the project it is difficult to tell what assumptions are incorrect until costly mistakes have been made. Requirements are defined and then that portion of the process is closed. Closed stages are not revisited. All requirements must be reviewed early and must be accurate. User definitions that are incomplete in the beginning cannot be changed or added later, after the requirements phase is closed. Anything missed early on causes complications later. Since it nearly impossible to think of every possible requirement without a working product, this model doesn’t work well for developing new products.

Waterfall Model (Questions and answers)

What is the purpose of this lesson?

This lesson is an overview of the waterfall model of software development. During this lesson you will learn how the waterfall lifecycle model of software development works, what the phases of this model are, and what advantages and disadvantages there are to using this model.

Where did the waterfall model of software development come from?

The waterfall lifecycle model was first developed by Winston Royce in 1970. Since then this model has become one of the most popular methods used in software development.

How does the waterfall model work?

The waterfall lifecycle model is pretty straightforward. This method of software development is set up in five sequential steps. During each step everything that is done, or that will need to be done, is documented. These
documents are critical to the success of this model. Each stage is closed before going to another stage and everything that has been done is locked and cannot be revisited once a stage is closed.

**What are the stages of the waterfall model?**

The stages of the waterfall model are; (1) requirements, (2) design, (3) implementation, (4) testing, and (5) maintenance. Each stage has a purpose and is gone through separately from the other phases.

**What is done in the requirements stage?**

During the requirements stage the client and developers decide upon requirements for the system being developed. These requirements are thoroughly documented. When this stage closes the only products you have are those documents.

**What is done in the design phase?**

The design is the phase in which all design details for low level and high level designs are decided upon. This is where you decide which programs are going to be used and how they are going to work together. All aspects of design are taken into consideration and documented. When this stage closes the only products you have are documents.

**What is done in the implementation phase?**

The implementation phase is where the design details are taken and made into a product. Coding starts in this phase. This is where your product comes to life. Before this stage is finished some small tests may be ran but for the most part this stage doesn’t include testing, it only includes writing code to build the software you have been planning.

**What is done in the testing phase?**

The testing phase is where you finally have something to work with. This is the phase that the client finally feels like you are getting somewhere. Life is being given to your software. During the testing phase you work the bugs out
of your new program. Once the software is ready for the client to assist they will join you with testing.

What is done in the maintenance phase?

After you are satisfied the software meets the requirements you have specified and it is working to the standards you wanted you enter the maintenance phase. This is where you do routine maintenance and update as needed. Your program remains in this phase when it is finished. If you need to change something or update the software you begin at the requirements phase again.

When should I choose to use the waterfall model?

The waterfall model works best when familiar software is being created. If you are updating old software or if you are making a program that works like an existing program the waterfall method works well.

When should I choose not to use the waterfall model?

The waterfall model doesn’t work well in situations where you have little information to begin with. When you are creating software that functions in a way like no other programs or when you are not familiar with the programs your software mimics you will have difficulties with the waterfall method. When the requirements are vague you shouldn’t use this method. This is because of the need to have all of the information at the beginning of the process, during the requirements phase.

What are the strong points of the waterfall model?

With this model of software development the stages are well defined. Everyone knows what they are supposed to be doing and when they are supposed to do it. Managers are able to easily oversee a project and know if it is on track or not by the documentation. Documentation along the way should be done well enough that developers are able to join the team late in the development.

What are the negative points of the waterfall model?
The waterfall model assumes that software development is a straight, step by step progression from beginning to end without looking back. In reality changes often need to be made, requirements change, and details in the beginning are often missing key information. User definitions during the requirement stage that are incomplete can’t be updated after the stage is closed. This model has no way to go back and make changes after a stage is finished and locked.

Working through the first stages with nothing but documents to show for it can be frustrating. Clients often get worried when they don’t see results during the early stages of development when the only product is documentation.
Appendices
Stages or Phases of the Waterfall Model

The model begins with the requirements stage. At this stage, the system requirements are defined and documented during the analysis phase.

During the design phase, the detailed design is created. Additionally, the system architecture is defined. The design phase involves the creation of detailed models and specifications that guide the development process.

The implementation phase involves the actual coding of the software. The code is written based on the designs created during the previous phase.

The testing phase involves the testing of the software to ensure it meets the requirements defined during the earlier phases. This phase includes various types of testing such as unit testing, integration testing, system testing, and acceptance testing.

The deployment phase involves the installation and deployment of the software to the production environment. After the software is deployed, it goes through a period of stabilization and acceptance testing.

The maintenance phase involves the ongoing support and improvement of the software. During this phase, bugs are fixed, features are added, and performance is optimized.

Press and Cons of the Waterfall Model

Pros:

- Clear and structured process
- Linear progression from one stage to the next
- Easy to manage and control

Cons:

- Lack of flexibility
- Risk of project failure if requirements change late in the process
- Potential for high rework costs if changes are needed late in the process
Appendices
Appendices
APPENDIX D
PRE-TEST / POST-TEST
Pre-test / post-test#

Please press Save button after you answer.

Q1- In waterfall life cycle, software development goes through various steps:
1- discovering errors, design, maintenance, publish, requirements and implementation.
2- requirements, incremental model, implementation, maintenance, and design.
3- requirements, design, implementation, testing, and maintenance.
4- design, testing, maintenance, requirements, and implementation.
5 - implementation, design, testing, maintenance, and requirements.

Q2- The waterfall model has a unique way of showing progress in stages. Stages are set in ......
1- a spiral manner.
2- a linear manner.
3- a hyperlink manner.
4- a hierarchical manner.
5- a cyclic manner.

Q3- In the pure Waterfall model , ..... 
1- every stage starts with design
2- all stages finish at the same time.
3- you work on one stage at a time.
4- you work on two stages at once.
5- you work on each stage more than once.

Q4- In waterfall model, software is developed ....
1- after the design stage.
2- at the beginning of the plan.
3- at the end of each stage.
4- after the testing stage.
5- before the design stage.
Q5- The design stage is where you:
1- write the code.
2- decide how the program will work.
3- gather the requirements.
4- test the system.
5- implement the software.

Q6- Writing the code is called the........ stage.
1- design
2- implementation
3- requirements
4- integration
5- verification

Q7- When does testing stage begin?
1- immediately after the requirement stage
2- immediately after the maintenance stage
3- immediately before the requirement stage
4- immediately after the implementation stage
5- immediately after the design stage

Q8- What is the last stage of the waterfall model?
1- design stage.
2- maintenance stage
3- testing stage
4- verification stage
5- integration stage
Please press SAVE button after you answer the following questions.

Q1- In waterfall life cycle, software development goes through various steps:
1- discovering errors, design, maintenance, publish, requirements, and implementation.
2- requirements, incremental model, implementation, maintenance, and design.
3- requirements, design, implementation, testing, and maintenance.
4- design, testing, maintenance, requirements, and implementation.
5- implementation, design, testing, maintenance, and requirements.

Q2- The waterfall model has a unique way of showing progress in stages. Stages are set in ...
1- a spiral manner.
2- a linear manner.
3- a hyperlink manner.
4- a hierarchical manner.
5- a cyclic manner.

Q3- In the pure Waterfall model, ..... 
1- every stage starts with design
2- all stages finish at the same time.
3- you work on one stage at a time.
4- you work on two stages at once.
5- you work on each stage more than once.

Q4- In waterfall model, software is developed :  
1- after the design stage.
2- at the beginning of the plan.
3- at the end of each stage.
4- after the testing stage.
5- before the design stage.

Q5- The design stage is where you…. :
1- write the code.
2- decide how the program will work.
3- gather the requirements.
4- test the system.
5- implement the software.

Q6- Writing the code is called the……… stage. :
1- design
2- implementation
3- requirements
4- integration
5- verification

Q7- When does testing stage begin? :
1- immediately after the requirement stage
2- immediately after the maintenance stage
3- immediately before the requirement stage
4- immediately after the implementation stage
5- immediately after the design stage

Q8- What is the last stage of the waterfall model? :
1- design stage.
2- maintenance stage
3- testing stage
4- verification stage
5- integration stage
Q9 - Which of the following are considered to be advantages of the waterfall model?

1- It has a linear approach with well defined stages.

2- It is simple to follow.

3- It allows you to iterate through stages.

4- It allows you to create a prototype early on.

5- It allows you to move freely between stages.

Q10 - All requirements must be

1- updated throughout the whole process

2- reviewed and finalised early in the process

3- checked at the end of the whole process.

4- altered after testing the system.

5- extracted from the implementation.

Q11: In waterfall life cycle, software development goes through number of steps

1- requirements, design, implementation, testing, and maintenance.

2- requirements, incremental model, implementation, maintenance, and design.

3- implementation, design, testing, maintenance, and requirements.

4- design, testing, maintenance, requirements, and implementation.

5- discovering errors, design, maintenance, publish, requirements, and implementation.

Q12 - The maintenance stage is where... :

1- all requirements should be maintained early.

2- testing is accomplished fast.

3- improvement of the software is done once it is tested.

4- the system is separated into smaller parts.

5- the design is re-architected

Q13 - Which of the following are considered to be disadvantages of the waterfall model? :

1- The process is not well defined.

2- It is complex to use.
3- It is possible to combine two stages at the same time.
4- A lack of iteration.
5- A lack of focus on security issues

Q14 - ........................affects the difficulty level of coding during the implementation stage. :

1- The testing stage...
2- The documentation quality during the design stage...
3- The deployment of the system...
4- The stages of the waterfall model ...
5- The maintenance stage...

Q15- When was the Waterfall model invented? :

1- 2003.
3- 1990.
4- 1970.
5- 1996.

Q16- At which point in the process will the requirements be finalised?

1- after finishing the system
2- during implementation
3- before starting the system
4- after the design stage
5- at any time.

Q15 A- Who invented the Waterfall model? :

1- Rational Software.
2- Barry Boehm.
3- Winston Royce.
4- DSDM Consortium
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I would recommend this system to my colleagues.</td>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
</tr>
<tr>
<td>2. The instructions and prompts are clear.</td>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
</tr>
<tr>
<td>3. Learning to operate this software is easy.</td>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
</tr>
<tr>
<td>4. I enjoyed my session with this system.</td>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
</tr>
<tr>
<td>5. I find that the system is useful for learning.</td>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
</tr>
<tr>
<td>6. Working with this software is satisfying.</td>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
</tr>
<tr>
<td>7. The way that system information is presented is clear and understandable.</td>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
</tr>
<tr>
<td>8. The organisation of the system seems logical.</td>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
</tr>
<tr>
<td>9. The software has an attractive presentation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
<td>Strongly agree</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
<td>-----------</td>
<td>-------</td>
<td>----------------</td>
</tr>
<tr>
<td>10- It is easy to move from one part to another.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
<td>Strongly agree</td>
</tr>
<tr>
<td>11- I have a better understanding of the topic “waterfall lifecycle” by using this lesson.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
<td>Strongly agree</td>
</tr>
<tr>
<td>12- I would use this system again.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly disagree</td>
<td>Disagree</td>
<td>Undecided</td>
<td>Agree</td>
<td>Strongly agree</td>
</tr>
</tbody>
</table>
APPENDIX E
SCREENSHOTS
Adaptive system to learning styles

Welcome, alaa

Thank you for entering the Adaptive System to Learning Styles

You will need to complete the site in the following order: (1) the VARK test, (2) the Honey and Mumford test, (3) Leaone, and finally (4) a feedback questionnaire.

STUDENT PROFILE

<table>
<thead>
<tr>
<th>VARK</th>
<th>Visual</th>
<th>Audial</th>
<th>Read Write</th>
<th>Kinesthetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

TESTS

- VARK TEST
- Honey & Mumford TEST

You haven't taken Honey & Mumford test

This work is based on an ongoing research project School of Mathematical and Computer Sciences, UK

Adaptive system to learning styles

Welcome, alaa

Thank you for entering the Adaptive System to Learning Styles

You will need to complete the site in the following order: (1) the VARK test, (2) the Honey and Mumford test, (3) Leaone, and finally (4) a feedback questionnaire.

STUDENT PROFILE

<table>
<thead>
<tr>
<th>VARK</th>
<th>Visual</th>
<th>Audial</th>
<th>Read Write</th>
<th>Kinesthetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

TESTS

- VARK TEST
- Honey & Mumford TEST

You haven't taken Honey & Mumford test

This work is based on an ongoing research project School of Mathematical and Computer Sciences, UK

Adaptive system to learning styles

Welcome, alaa

Thank you for entering the Adaptive System to Learning Styles

You will need to complete the site in the following order: (1) the VARK test, (2) the Honey and Mumford test, (3) Leaone, and finally (4) a feedback questionnaire.

STUDENT PROFILE

<table>
<thead>
<tr>
<th>VARK</th>
<th>Visual</th>
<th>Audial</th>
<th>Read Write</th>
<th>Kinesthetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

TESTS

- VARK TEST
- Honey & Mumford TEST

You haven't taken Honey & Mumford test

This work is based on an ongoing research project School of Mathematical and Computer Sciences, UK

Appendices
Adaptive system to learning styles

Thank you for entering the Adaptive System to Learning Styles

You will need to complete the site in the following order: (1) the VARK test, (2) the Honey and Mumford test, (3) Lessons, and finally (4) a feedback questionnaire.

<table>
<thead>
<tr>
<th>STUDENT PROFILE</th>
<th>TESTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vark</td>
<td></td>
</tr>
<tr>
<td>Visual</td>
<td>7</td>
</tr>
<tr>
<td>Auditory</td>
<td>3</td>
</tr>
<tr>
<td>Read/Write</td>
<td>0</td>
</tr>
<tr>
<td>Kinesthetic</td>
<td>0</td>
</tr>
<tr>
<td>Honey and Mumford</td>
<td>5</td>
</tr>
<tr>
<td>Abstract</td>
<td>9</td>
</tr>
<tr>
<td>Reflective</td>
<td>8</td>
</tr>
<tr>
<td>Theoretical</td>
<td>5</td>
</tr>
<tr>
<td>Pragmatic</td>
<td>4</td>
</tr>
</tbody>
</table>

This work is based on an ongoing research project School of Mathematical and Computer Sciences, UK.

Adaptive system to learning styles

Waterfall Lifecycle

Objectives 1/5

Overview

This lesson introduces students to the Waterfall life cycle model. On completion students will have an understanding of the waterfall model including concept, phases, advantages, disadvantages, and when to use the waterfall model cycle model.

Other Learning Styles

Vark: Visual, Read/Write, Kinesthetic

Honey and Mumford: Abstract, Reflective, Theoretical, Pragmatic