1. **What is an Activity Diagram What are its purpose?**

**Answer:** *Activity diagram* can be used to model different aspects of a system. At a high level, they can be used to model business activates in an existing or potential system.

It can be used for the following purpose:

1. To model a task;
2. To describe a system function that is represented by a use case;
3. In operation specifications, to describe the logic of an operation.
4. **What is the difference between model and diagram?**

**Answer:**

|  |  |
| --- | --- |
| **model** | **diagram** |
| 1. *Models* represent system at different levels of details, which contains elements such as actors, use cases, classes, and packages. | 1. *Diagrams* are used to build models of system in the systems, in the same way as architects use drawings and diagrams to model buildings. |
| 2. A model can evolve as we learn more about a task or a problem. | 2. It can generate new ideas and possibilities. |

1. **How does a collaboration diagram differ from class diagram?**

**Answer:** A *collaboration diagram* shows only those classes that collaborate to provide the functionality of a particular use cases (or operations) and the links that are shown, are those that are required for that purpose.

A *class diagram* typically shows all the classes in a particular package and all the associations between them.

1. **What do you mean by Prototyping? What are the steps to prepare prototype?**

**Answer:** In software development a *prototype* is a system or a partially completed system, which is built quickly to explore some aspect of a system requirements and that is not intended as the final working system.

Main system requires the steps to prepare prototype:

1. Perform an initial analysis;
2. Define prototype objectives;
3. Specify prototype;
4. Construct prototype;
5. Evaluate prototype and recommend change.
6. **Difference between IS and IT.**

**Answer:**

|  |  |
| --- | --- |
| **Information system(IS)** | **Information Technology(IT)** |
| *IS* played an important part of human affairs, which is used to capture, store, organize and display information. | *IT* Strategy is responsible for identifying the hardware components and configurations that will allow the software to operate effectively. |
| It only be considered in the context of well thought-out business strategy. | Itonly be considered in the context of specific information systems that are planned for development. |
| *IS* strategy is about, what is feasible. | The role of *IT* strategy is to enable the successful defined in the *information system strategy*. |

1. **Difference between feedback and feed forward.**

**Answer:**

|  |  |
| --- | --- |
| **Feedback** | **Feed forward** |
| *Feedback* is sampling one or more outputs of a system for comparison to a control value. | *Feed forward* is sampling a system input, usually before it enters the system. |
| No one can develop a new skill, without receiving appropriate feedback. | Feed forward control information can help a system to be more responsive to environmental fluctuations. |

1. **What is the UML notation for each of the following package, sub-system and model?**

**Answer:**

Use case

Model

Campaign

Management

Use Cases

Package

Model

Sub-system

Fig: UML notation for packages, sub-systems and models.

1. **Draw a simple Activity diagram.**

**Answer:**

Fig: *Activity diagram* for the Activity write chapter.

(Satisfied)

(Not Satisfied)

Add Re-exercise

To Bibliography

Add Exercise

Revise Draft

Produce First Draft

Plan Chapter

1. **List the name of the fact finding techniques.**

**Answer:** There are 5 main *fact finding techniques* that are used by *analyst* to investigate requirements-

1. Background reading;
2. Interviewing;
3. Observation;
4. Document Sampling;
5. Questionnaires.
6. **Distinguish between composition from aggregation.**

**Answer:**

|  |  |
| --- | --- |
| **Composition** | **Aggregation** |
| 1. *Composition* is a type of abstraction that encapsulates groups of classes that collectively have the capacity to be a reusable sub-assembly, which represent the whole and the other part of the whole. | 1. *Aggregation* represents a whole part association between two or more objects. |
| 1. Symbol | 1. Symbol |
| 1. A part can belong only one composition. | 1. A part can belong more than one aggregation. |

1. **Difference between sequence diagram and collaboration diagram.**

**Answer:**

|  |  |
| --- | --- |
| **Sequence diagram** | **Collaboration diagram** |
| 1. *Sequence diagram* shows an interaction between objects arranged in a time sequence. | 1. *Collaboration diagram* shows an interaction between objects and the contents of the interaction in terms of the links between the objects. |
| 1. It has a time dimension. | 1. It doesn’t has time dimension. |
| 1. It does not show the link between objects. | 1. It shows the link between objects. |

1. **What are the basic elements of UML model diagram?**

**Answer:***UML diagrams* are made up of four elements-

1. Icons;
2. Two dimensional symbols;
3. Paths;
4. Strings.
5. **Phases of waterfall life cycle.**

**Answer:** Phases of waterfall life cycle are given below:

* System engineering;
* Requirements analysis;
* Design;
* Construction;
* Testing;
* Installation;
* Maintenance.

1. **What is a collaboration diagram?**

**Answer:** A *collaboration diagram* shows an interaction between objects and the context of the interaction in terms of the link between the objects.

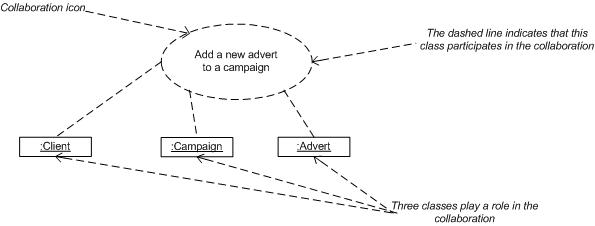


Fig: Collaboration for add a new advert to a campaign.

1. **Difference between algorithmic and non-algorithmic technique to operation specification.**

**Answer:**

|  |  |
| --- | --- |
| **Algorithmic technique** | **Non-Algorithmic technique** |
| 1. *An algorithm* *technique*defines the step-by-step behavior of an operation. | 1. *A non-algorithmic technique* approach defines only inputs and results. |
| 1. It also specifies the sequence in which the steps are performed. | 1. It does not specifies the sequence. |
| 1. Generally it does not prefer in object-oriented development. | 1. Generally it is preferred in object-oriented development. |
| 1. It describes the internal logic. For example: *Activity Diagram.* | 1. It does not describe the internal logic. For example: *Decision table*. |