

COMPUTER

CHAPTER – 3

ELEMENTRY CONCEPTS OF OBJECTS AND CLASSES

QUESTIONS & ANSWERS

Q1: what are objects? Give five examples.

Ans: when you look around, you will see objects everywhere. The chair you are sitting on, the table, pen, book, fan, door, etc.

1. **Tangible Objects:** These are the objects which you can see and touch. For example, chair and computer. In fact you are also an object.
2. **Conceptual Objects:** These are intangible objects and exist as a conceptual entity. You cannot see and touch them for example, an email or a bank account.
3. **Roles:** roles played by people like student, a teacher or a clerk.
4. **Event:** an event is something occurring in a system. For example, a sale or purchase in a departmental store. Objects are the basic units of an object-oriented system, an object is an identifiable entity that has its own set of attribute, behaviour and state.

Q2: explain the anatomy of an object.

Ans: The new keyword is used to create an object of the class type. The new keyword allocates memory for the new object.

Q3: explain the following statement “Class is a specification for Objects”.

Ans: A class is just a specification of the object. The attributes and methods in a class are thus declarations that do not contain any values. However, an object is a concrete instance of a class with properly defined values of each attributes and behaves as per the methods of the class.

Q4: Explain the following statement “Abstraction is relative to the perspective of the viewer”

Ans: Driver’s View: for the driver, a car is one single object, which can be operated using gears, indicators, accelerator, break, etc. the driver doesn’t need to know or understand the internal mechanisms of how these components actually work.

Mechanic’s view: Car mechanics need to know and understand how the internal mechanism like gears, indicators, accelerator, brake, etc. actually work so that they can identify and fix issues with the cars, for a mechanic, each car part is a single objects.

Q5: identify five possible attributes and their possible behaviours of the following entities:

- a) **School** – Attribute
- b) **Student** – Attribute
- c) **Teacher** – Attribute
- d) **Computer** – Behaviour
- e) **Television** – Behaviour
- f) **Washing machine** – Behaviour

g) Chair – Attribute

h) Person – Attribute

Q6. Explain in details how a class is different form an object.

Ans: a class is just a specification of the object. The attributes and methods in a class are thus declarations that do not contain any values. However, an object is a concrete instance of ta class with properly defined values of each attribute and behaves as per the methods of the class.

Q7. Give the reason why a class is known as:

a) An object factory: a class is called an object factory because objects are created from the class that contains common attributes and behaviours. The class behaves like a specification for creating such similar objects.

b) A composite data type: A class is composed of member variables which are of different data types. Hence, a class can be viewed as a composite data type.

c) A user-defined data type: The entire data and the code, contained in a object, becomes a user-defined data type using concept of a class the class may be considered as a data type and an object as a variable of that data type. For example once the bird class has been defined, the statement.

Bird, Parrot;

Q8: Cancelled

Q9: What do you understand by the term objects encapsulate state and behaviour?

Ans: An object stores its state in member variables and exposes its behaviour through the member methods. The member methods operate on member variables and serve as the primary mechanism to interact with the object. Only the member methods which are wrapped inside the class can access the data and change its state. Hence, the state and behaviour are said to encapsulate by the object, hiding internal state and requiring all interaction to be performed through an object's methods.

Q10: Explain how objects communicate with each other.

Ans: Objects communicate with each other by sending messages. The sender object requests the receiver object to perform an action.