**PART-A 15 MARKS**

**Instructions: Common to all SETS**

1. Define the terms class, object, encapsulation, Polymorphism, Abstraction, constructor. **6M**
2. Why main() method is public, static and void in java. **3M**
3. Explain about JDK, JRE and JVM. **6M**

**PART-B 35 MARKS**

**Instructions: For each case study given in each SET:**

Write a JAVA program.

Find suitable datatypes, access specifiers.

Use **fina**l, **static** and **this** keyword as per requirement.

Display a message “Welcome to PVPSIT” such that it executes before main method.

Create instances with different constructors.

Perform operations

Display details to verify operations.

**SET-A**

**Define a class Account with the following attributes and methods:**

**Attributes:**

bankName (String=”PVPSIT”) is same for all accounts and cannot be altered

accountNumber (int)

balance (double)

**Methods:**

• Account(int accountNumber, double balance): Parameterized constructor to initialize attributes.

• Copy Constructor: Takes another Account object as a parameter and initializes a new Account object with the same property values as the passed object.

• getAccountNumber(), setAccountNumber(int accountNumber): Getter and setter methods for accountNumber.

• getBalance(), setBalance(double balance): Getter and setter methods for balance.

• deposit(double amount): Method to add amount to the balance.

• withdraw(double amount): Method to subtract amount from the balance (considering validation for sufficient funds).

• displayAccountDetails(): Method to display account details (bankName, accountNumber and balance).

**SET-B**

**Define a class Employee with the following attributes and methods:**

**Attributes:**

organisation (String=”PVPSIT”) is same for all employees and cannot be altered

name (String),

age (int),

designation (String).

**Methods:**

• Employee(String name, int age, String designation): Parameterized constructor to initialize attributes.

• Copy Constructor: Takes another Employee object as a parameter and initializes a new Employee object with the same property values as the passed object.

• getName(), setName(String name): Getter and setter methods for name.

• getAge(), setAge(int age): Getter and setter methods for age.

• getDesignation(), setDesignation(String designation): Getter and setter methods for designation.

• displayDetails(): Method to display details of the employee.

**SET-C**

**Define a class Circle with attributes and methods:**

**Attributes:**

radius (double).

color (String)

Filled (Boolean)

**Methods:**

• Circle(): Default constructor initializing radius to 1.0.

• Circle(double radius): Parameterized constructor.

• Circle(double radius, String color, boolean filled): Parameterized constructor

• Copy Constructor: Takes another Circle object as a parameter and initializes a new Circle object with the same property values as the passed object.

• getRadius(): Getter method for radius.

• setRadius(double radius): Setter method for radius.

• getArea(): Method to calculate and return the area of the circle (area = π \* radius \* radius).

• getPerimeter(): Method to calculate and return the perimeter of the circle (perimeter = 2 \* π \* radius).

**SET-D**

**Define a class Shape with the following attributes and methods:**

**Attributes:**

color (String), filled (boolean).

**Methods:**

• Shape(): Default constructor initializing color to "green" and filled to true.

• Shape(String color, boolean filled): Parameterized constructor.

• Copy Constructor: Takes another Shape object as a parameter and initializes a new Shape object with the same property values as the passed object.

• getColor(): Getter method for color.

• setColor(String color): Setter method for color.

• isFilled(): Method to check if the shape is filled.

• setFilled(boolean filled): Method to set whether the shape is filled.

**SET-E**

**Define a class Product with the following attributes and methods:**

**Attributes:**

companyName (String=”PVPSIT”) is same for all products and cannot be altered

productId: Unique identifier for each product (int).

productName: Name of the product (String).

price: Price of the product (double).

quantity: Quantity of the product in stock (int).

**Methods:**

• Constructor with four parameters (productId, productName, price, quantity).

• Constructor with three parameters (productId, productName, price), initializing quantity to 0.

• Constructor with two parameters (productId, productName), initializing price to 0.0 and quantity to 0.

• Copy Constructor: Takes another Product object as a parameter and initializes a new Product object with the same property values as the passed object.

• addStock(int quantity): Increases the stock quantity of the product.

• sellStock(int quantity): Decreases the stock quantity of the product when sold.

• sellStock(int quantity, double discount): Decreases the stock quantity of the product when sold with a discount, updating the price accordingly.

• displayProductDetails(): Displays details of the product including companyName,productId, productName, price, and quantity.

**SET-F**

**Define a class Book with the following attributes and methods:**

**Attributes:**

publisherName (String=”PVPSIT”) is same for all books cannot be altered

title: Title of the book (String).

author: Author of the book (String).

isbn: ISBN (International Standard Book Number) of the book (String).

price: Price of the book (double).

**Methods:**

Constructor: You need to implement the following constructors:

• Default Constructor:

Initializes title to "Unknown Title".

Initializes author to "Unknown Author".

Initializes isbn to "000-0-00-000000-0".

Initializes price to 0.0.

• Parameterized Constructor:

Takes title, author, and isbn as parameters and initializes price to a default value of 20.0.

• Parameterized Constructor:

Takes title, author, isbn, and price as parameters and initializes all properties accordingly.

• Copy Constructor: Takes another Book object as a parameter and initializes a new Book object with the same property values as the passed object.

• updatePrice(double newPrice): that updates the price of the book.

• displayBookInfo(): that displays detailed information about the book including publishername,title, author, ISBN, and price.

\*\*\*\*\*\*THE END\*\*\*\*\*\*

\*\*\*\*\*\*WISU U ALL THE BEST\*\*\*\*\*