

1) Which of the following binary numbers is equal to the base-16 (hexadecimal) number B7?

- a) 10110111 b) 11001000 c) 11011000 d) 10100111 e) none of the answers are correct

2) Which of the instance variables in the `PizzaOrder` class below need to be declared as static for the class to correctly output the following when the main method is run?

Pizza size: small, Toppings: 3, Order number: 1, Number of pizzas: 3
Pizza size: medium, Toppings: 2, Order number: 2, Number of pizzas: 3
Pizza size: large, Toppings: 4, Order number: 3, Number of pizzas: 3

- a) `size` b) `toppings` c) `orderNumber` d) `numberOfPizzas`

```
public class PizzaOrder {

    private String size;
    private int toppings;
    private int orderNumber;
    private int numberOfPizzas;

    public PizzaOrder(String s, int t) {
        size = s;
        toppings = t;
        numberOfPizzas++;
        orderNumber = numberOfPizzas;
    }

    public String toString() {
        return "Pizza size: " + size + ", Toppings: " + toppings +
            ", Order number: " + orderNumber +
            ", Number of pizzas: " + numberOfPizzas;
    }

    public static void main(String args[]) {
        PizzaOrder p1 = new PizzaOrder("small",3);
        PizzaOrder p2 = new PizzaOrder("medium",2);
        PizzaOrder p3 = new PizzaOrder("large",4);

        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p3);
    }
}
```

Questions 3-5 refer to the following partial class definitions:

```
public class BankAccount {  
    private double balance;  
  
    public BankAccount(double amount) { balance = amount; }  
    public void deposit(double amount) { balance += amount; }  
    public void withdraw(double amount) { balance -= amount; }  
}
```

```
public class CheckingAccount extends BankAccount {  
    private double checkFee;  
  
    public CheckingAccount(double fee, double amount) {  
        // < method not complete >  
    }  
  
    public void withdraw(double amount) {  
        // < method not complete >  
    }  
    public void chargeCheckFee() {  
        // < implementation not shown >  
    }  
}
```

3) Complete the missing code in the `CheckingAccount` constructor.

4) Complete the missing code in the `withdraw` method, to only allow a withdrawal if the remaining balance is at least \$100.

5) The following code segments appear in a tester (client) class.

```
BankAccount a1 = new CheckingAccount (100,2);
```

Which statements below are valid?

```
a1.deposit(100.00);
```

```
a1.withdraw(10);
```

```
a1.balance -= 20;
```

```
a1.chargeCheckFee();
```