SET 2

1. Design a class BankAccount with properties accountNumber and balance, and methods deposit() and withdraw(). Extend this class with subclasses SavingsAccount and CheckingAccount. Implement specific rules such as minimum balance requirements and interest calculation for savings accounts.

Main.java

Share Run

Output

```
class BankAccount {
    protected String accountNumber;
    protected double balance;
    public BankAccount(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: $" + amount);
    }
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
        } else {
            System.out.println("Insufficient funds");
        }
    }
}
class SavingsAccount extends BankAccount {
    private double interestRate;
    public SavingsAccount(String accountNumber, double balance, double interestRate) {
        super(accountNumber, balance);
        this.interestRate = interestRate;
    }
    public void addInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest added: $" + interest);
    }
}
```

```
java -cp /tmp/JOkZrUswfZ/Main
Deposited: $500.0
Interest added: $37.5
Withdrawn: $200.0
Deposited: $300.0
Withdrawn: $2500.0

=== Code Execution Successful ===
```

Main.java

Share Run

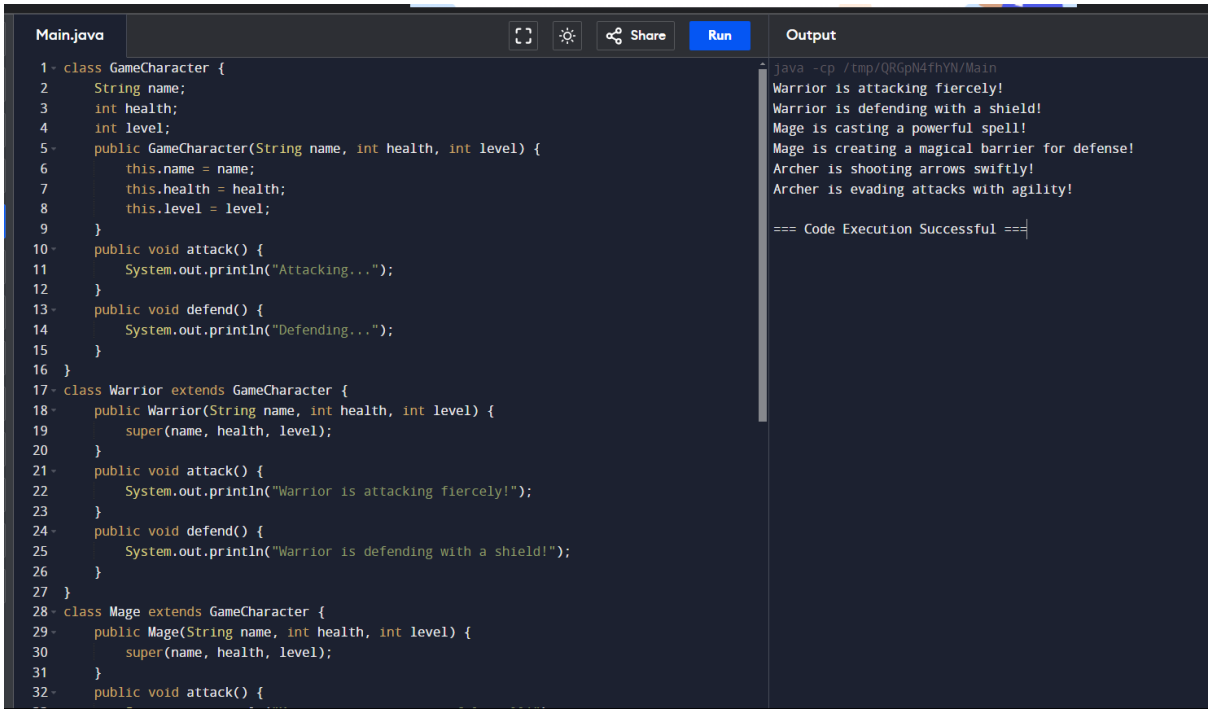
Output

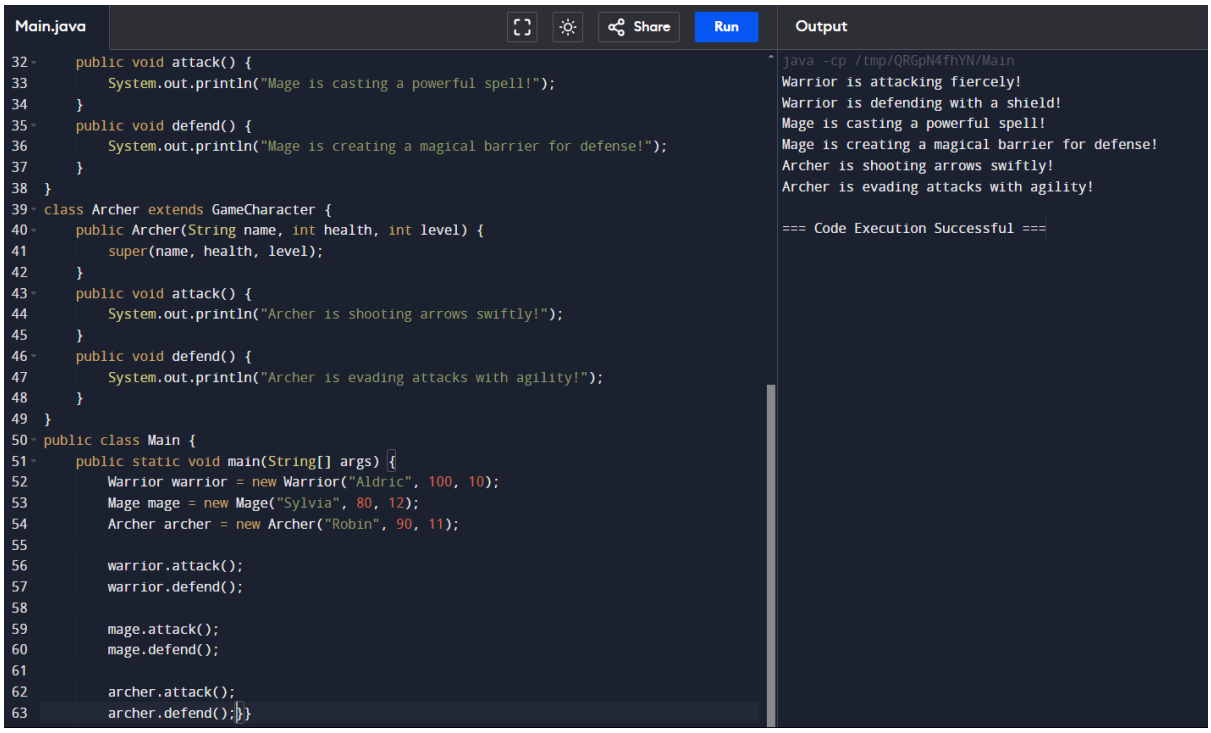
```
        balance += interest;
        System.out.println("Interest added: $" + interest);
    }
}
class CheckingAccount extends BankAccount {
    private double overdraftLimit;
    public CheckingAccount(String accountNumber, double balance, double overdraftLimit)
        {
        super(accountNumber, balance);
        this.overdraftLimit = overdraftLimit;
    }
    public void withdraw(double amount) {
        if (balance + overdraftLimit >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
        } else {
            System.out.println("Exceeded overdraft limit");
        }
    }
}
public class Main {
    public static void main(String[] args) {
        SavingsAccount savings = new SavingsAccount("SAV123", 1000.0, 2.5);
        savings.deposit(500.0);
        savings.addInterest();
        savings.withdraw(200.0);
        CheckingAccount checking = new CheckingAccount("CHK456", 2000.0, 500.0);
        checking.deposit(300.0);
        checking.withdraw(2500.0);
    }
}
```

```
java -cp /tmp/JOkZrUswfZ/Main
Deposited: $500.0
Interest added: $37.5
Withdrawn: $200.0
Deposited: $300.0
Withdrawn: $2500.0

=== Code Execution Successful ===
```

2. Create a base class GameCharacter with properties name, health, and level. Extend this class with subclasses Warrior, Mage, and Archer. Implement methods such as attack() and defend() differently for each subclass, showcasing polymorphism through inheritance.





3. Design a class Product with properties productId, name, and price. Extend this class with subclasses Electronics and Clothing. Implement methods to calculate discounts based on

membership status for electronics and seasonal sales for clothing.

Main.java

Share Run

```
class Product {
    int productId;
    String name;
    double price;

    public Product(int productId, String name, double price) {
        this.productId = productId;
        this.name = name;
        this.price = price;
    }
}

class Electronics extends Product {
    public Electronics(int productId, String name, double price) {
        super(productId, name, price);
    }

    public double calculateDiscount(double price, boolean isMember) {
        if (isMember) {
            return price * 0.1;
        } else {
            return 0.0;
        }
    }
}
class Clothing extends Product {
    public Clothing(int productId, String name, double price) {
        super(productId, name, price);
    }

    public double calculateDiscount(double price, boolean isSeasonSale) {
        if (isSeasonSale) {
```

Output

```
java -cp /tmp/9UemYHe0RL/Main
Electronic Product Discount: $100.0
Clothing Product Discount: $4.0

=== Code Execution Successful ===
```

Main.java

Share Run

```
}
class Clothing extends Product {
    public Clothing(int productId, String name, double price) {
        super(productId, name, price);
    }

    public double calculateDiscount(double price, boolean isSeasonSale) {
        if (isSeasonSale) {
            return price * 0.2;
        } else {
            return 0.0;
        }
    }
}
public class Main {
    public static void main(String[] args) {
        Electronics electronicProduct = new Electronics(1, "Laptop", 1000.0);
        Clothing clothingProduct = new Clothing(2, "T-shirt", 20.0);

        boolean isMember = true;
        boolean isSeasonSale = true;

        double electronicDiscount = electronicProduct.calculateDiscount
            (electronicProduct.price, isMember);
        double clothingDiscount = clothingProduct.calculateDiscount(clothingProduct
            .price, isSeasonSale);

        System.out.println("Electronic Product Discount: $" + electronicDiscount);
        System.out.println("Clothing Product Discount: $" + clothingDiscount);
    }
}
```

Output

```
java -cp /tmp/9UemYHe0RL/Main
Electronic Product Discount: $100.0
Clothing Product Discount: $4.0

=== Code Execution Successful ===
```

4. Design a class LibraryItem with properties title, author, and year. Extend this class with subclasses Book and DVD. Implement methods for checking in and out items, and display detailed information for each item type.

Main.java

Share Run

Output

```
class LibraryItem {
    String title;
    String author;
    int year;
    public LibraryItem(String title, String author, int year) {
        this.title = title;
        this.author = author;
        this.year = year;
    }
    public void checkOut() {
        System.out.println("Checking out: " + title);
    }
    public void checkIn() {
        System.out.println("Checking in: " + title);
    }
    public void displayInfo() {
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Year: " + year);
    }
}
class Book extends LibraryItem {
    public Book(String title, String author, int year) {
        super(title, author, year);
    }
}
class DVD extends LibraryItem {
    public DVD(String title, String author, int year) {
        super(title, author, year);
    }
}
lass Main {
```

```
java -cp /tmp/nztwhQa1VT/Main
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Year: 1925
Checking out: The Great Gatsby
Checking in: The Great Gatsby

Title: Inception
Author: Christopher Nolan
Year: 2010
Checking out: Inception
Checking in: Inception

=== Code Execution Successful ===
```


Main.java

Share Run

Output

```
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Year: " + year);
    }
}
class Book extends LibraryItem {
    public Book(String title, String author, int year) {
        super(title, author, year);
    }
}
class DVD extends LibraryItem {
    public DVD(String title, String author, int year) {
        super(title, author, year);
    }
}
public class Main {
    public static void main(String[] args) {
        Book book = new Book("The Great Gatsby", "F. Scott Fitzgerald", 1925);
        DVD dvd = new DVD("Inception", "Christopher Nolan", 2010);

        book.displayInfo();
        book.checkOut();
        book.checkIn();

        System.out.println();

        dvd.displayInfo();
        dvd.checkOut();
        dvd.checkIn();
    }
}
```

```
java -cp /tmp/nztwhQa1VT/Main
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Year: 1925
Checking out: The Great Gatsby
Checking in: The Great Gatsby

Title: Inception
Author: Christopher Nolan
Year: 2010
Checking out: Inception
Checking in: Inception

=== Code Execution Successful ===
```