1  1. class Parameters {
2.     /*
3.     * An entry point to execution
4.     */
5.     public static void main(String[] args) {
6.         if (args.length <= 0) return;
7.         for (int i = 0; i < args.length; i++)
8.             System.out.println(i + " ". + args[i]);
9.     }
10. }

2  2. (Comments)

1. class Parameters {
2.     /*
3.     * An entry point to execution
4.     */
5.     public static void main(String[] args) {
6.         if (args.length <= 0) return;
7.         for (int i = 0; i < args.length; i++)
8.             System.out.println(i + " ". + args[i]);
9.     }
10. }

11. class Parameters {
12.     /*
13.     * An entry point to execution
14.     */
15.     public static void main(String[] args) {
16.         if (args.length <= 0) return;
17.         for (int i = 0; i < args.length; i++)
18.             System.out.println(i + " ". + args[i]);
19.     }
20. }
3  

```java
class Account {
    private double balance = 0;

    void deposit(double d) {
        balance += (d <= 0 ? 0 : d);
    }

    void withdraw(double w) {
        if (w <= balance) 
            balance -= w;
    }

    public double getBalance() {
        return balance;
    }

    public static void main(String[] args) {
        Account checking = new Account();
        double cash = 1024.48;
        checking.deposit(cash);
        System.out.println("balance=\n" + checking.getBalance());
        checking.withdraw(cash);
        System.out.println("balance=\n" + checking.getBalance());
    }
}
```
class Account {...}

f

2: (variables) java

4-6: (function) java

8-11: java

13-15: java

17-25: java

• eclipse.exe

• C:

• commands: java

1. eclipse.exe
2. `main` method should display the amount of money on the current balance. Java
   should launch Next button Finish.kwh

3. As `main` method should display the amount of money on the current balance. Java
   should launch Next button Finish.kwh

- **Example:**
  ```java
  class Account {
      private static int balance = 1024.48;
  }

  public class JavaApplication {
      public static void main(String[] args) {
          System.out.println(Account.balance);
      }
  }
  ```

4. **Example:**
   ```java
   public class Account {
       private static int balance = 1024.48;
   }

   public class JavaApplication {
       public static void main(String[] args) {
           System.out.println(Account.balance);
           Account balance = Account.balance;
       }
   }
   ```

   balance=1024.48
   balance=0.0
• **Static Initialization:** Static fields are initialized by the compiler. A static field is a memory location that is shared by all instances of a class. It is not automatically initialized when an object is created. Instead, the static initialization occurs when the class is loaded into memory. This allows the static field to be initialized before any of the object's methods are called.

- **Main Method:** The main method is the entry point of a Java program. It is the first method to be executed when the program starts. The main method typically contains the code that sets up the program and starts its execution.

5. **Static Initialization**

The static initialization occurs at runtime and is called before the constructor is called. The static initialization is used to initialize static fields, which are shared among all instances of a class. The static initialization is called before the constructor of the class is called, so it can be used to set up the class or its properties.

After the static initialization, the constructor is called to initialize the instance variables of the class. The constructor is used to initialize the object's properties, and it is called whenever an object is created.