**21CS203 – OBJECT ORINTED PROGRAMMING**

**ASSIGNMENT-1**

**SUBMITTED BY**

D JHANSI

21CSE073

**SUBMITTED TO**

J SANTHALAKSHMI REVATHY

DEPARTMENT OF COMPUTER SCIENCE

VELAMMAL COLLEGE OF

ENGINEERING AND TECHNOLOGY

|  |  |
| --- | --- |
| Content **(14)** |  |
| References **(2)** |  |
| Presentation **(2)** |  |
| Timely submission **(2)** |  |
| **Total (20)** |  |

**1. Relate oops concepts with real time examples**

**OBJECT ORIENTED PROGRAMMING**

It refers to programming languages that make use of objects. Inheritance, hiding, polymorphism, and other real-world concepts are all part of object-oriented programming. The basic goal of oops concepts with real time examples is to connect data and the functions that operate on it so that no other part of the code may access it except than function.

**CLASS**

The class is at the core of java. It is the logical construct upon which the entire java language is built because it defines the shape and nature of an object. As such, the class forms the basis for object-oriented programming in java. Any concept is implemented in a java programming must be encapsulated within a class.

**Example:**

A person is a simple example. The person class would have characteristics that represent data like the person’s age, name, height, and so on. The class description might also include functions like “saymyname” which prints that person’s name to the screen.

**OBJECTS**

It refers to real-life entity that is defined as an instance of a class. Each objects has its own identity, attribute and behavior. It take up memory and have an associated address. Each objects contains data as well as code for manipulating it.

**Example:**

In person class family could be created by creating individual person objects from the class for each family member. Because each person is different, each person’s object would have different data attributes.

**ABSTRACTION**

In abstraction, it displays only the important information by hiding the implementation parts.

**Example:**

Car is a set of tens of thousands of individual parts. They are well defined objects with its own unique behavior. This abstraction allows people to use a car to drive to the grocery store without being overwhelmed by the complexity of the parts that form the car. They can ignore the details of how the engine, transmission=on and braking system work. Instead they are free to utilize the object as a whole.

**ENCAPSULATION**

It is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse. Encapsulation is as a protective wrapper that prevents the code and data from being arbitrarily accessed by other code defined outside the wrapper. Access to the code and data inside the wrapper is tightly controlled through a well-defined interface.

**Example:**

Consider automatic transmission on an automobile. It encapsulates hundreds of bits of information about engine, such as hoe much you are accelerating, the pitch of the surface you are on, and the position of the shift lever. The transmission can’t affect by using the turn signal or windshield wipers. What occurs inside the transmission does not affect objects outside the transmission. Due to encapsulation dozens of car manufactures can implement one in any way.

The power of encapsulated code is that everyone know how to access it and thus can use it regardless of the implementation details – and without fear of unexpected side effects.

**INHERITANCE**

It allows classes to inherit common properties from the parent class. It is the process by which one object acquires the properties of another object. This is important because it supports the concept of hierarchical classification.

**Example:**

Let us take a class as animals. It inherits common properties like sound, color, breed from animal class and create classes like dog, cat, horse, etc. A golden retriever is part of the classification dog, which in turn is part of the mammal class under the larger class animal.

**POLYMORPHISM**

It is the ability to exit in many forms. It is a feature that allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation.

**Example:**

Boy as a real-world example. This boy can be a student, a player, and a writer. This boy can exist in different ways in different situations.

**2. What does the framework mean in java? Explain in detail any one java framework.**

**FRAMEWORK**

Java framework is the body or platform of prewritten codes used by java developers to develop java applications or web applications. It is a collection of predefined classes and functions that is used to process input, manage hardware devices interacts with system software. It acts like a skeleton by writing their own code.

Frameworks are the bodies that contain the prewritten codes in which code is added to overcome the problem. It use the programmer’s code because the framework is in control of the programmer. Framework is used by calling its methods, inheritance, and supplying callbacks, listeners, or other implementation of the observer pattern.

Some of the most popular java frameworks are:

* Spring
* Hibernate
* Grails
* Play
* JavaServer faces(JSF)
* Google Web Toolkit(GWT)
* Quarkus

**GOOGLE WEB TOOLKIT(GWT)**

It is an open-source framework that allows developers to write client-side java code. With the help of GWT, complex browse applications develop rapidly.

GWT is a development toolkit which enables developer to develop internet or web application. GWT insists on reusable approaches to common web-app tasks such as bookmarking, UI abstraction, cross-browser portability etc. It provides features such as:

* Developer need not to be expert at browser incompatibilities as GWT handles through different binding.
* GWT is a single code base for client and server.
* GWT has chosen Java with which development becomes more effective as it provide code navigation, refactoring etc.
* GWT MVP (Model View Presenter) allows shared working, faster testing with JUnit. The event bus allows many updates in client side application by placing event on the event bus.
* It allows different Java tools to integrate with it such as hibernate through gilead.

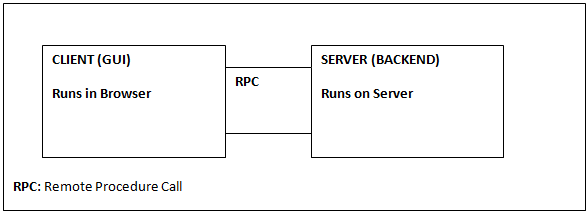
**COMPONENTS OF GWT**

* **GWT Java-to-JavaScript Compiler:** It translates all the Java written code into JavaScript.
* **GWT Development Mode:** It allows the developers to run application in development mode i.e. app run in Java without compiling JavaScript. Development mode allows native mode plug-in called Google Web Toolkit Developer Plug-in.
* **JRE emulation library:** GWT includes a library that follows the Java runtime library i.e. java.lang, java.lang.annotation, java.math, java.io, java.sql, java.util etc.
* **GWT Web UI class library:** For creating a widget GWT consists of set of interface and classes.

**WHY TO USE GWT?**

* **Performance Gain:** In GWT with each new release of the framework developer would be able to get all new features for free. Old files gain new features by just re-compiling such as accessibility support in GWT 1.5.
* **Debugging:** We can debug GWT apps just like in Java application, using IDE's Debugger.
* **OOPHM:** Out Of Process Hosted Mode(OOPHM) which allows developer to use hosted mode on the different browsers. It allows developer to increase GWT potential such as running Windows in VM, connecting to IE then running on the Hosted Mode.
* **UI Binder:** It allows you to design and customize your widgets in an easy and less complex way using XML, HTML, and Java.
* GWT takes a strong OO(Object Oriented) approach to Software Architecture.

**BREAKDOWN OF GWT APP**



**GWT PROVIDES TWO MODES**

1. **Development Mode:** It provides the facility to debug Java code of the application directly with your Java debugger.
2. **Web Mode:** This mode enables the application to be translated into HTML, CSS and JavaScript code which can be directly deployed to a web server.

**GWT INSTALLATION**

1. Download JDK
2. Install JDK and JRE
3. Set the path
4. Verify the JDK installation
5. Install Eclipse
6. Install GWT SDK and PLUGINS for Eclipse
7. Setup Apache

**ADVANTAGES**

* It employs reusability for web application development
* We can use google API to develop application.
* It provides functionality such as, internationalization, UI abstraction, and history management.

**APPLICATION DEVOLOPMENT PROCESS**

|  |  |  |
| --- | --- | --- |
| **TASKS** | **CONCEPTS** | **GWT TOOLS AND API** |
| 1. Create a GWT Project | Generate the Files and Directories. | Google Plugin for Eclipse, GWT command-line tool, webAppCreator in Development Mode. |
| 2. Design the application | Basic layout of the application. | Language Constraint. |
| 3. Build the User Interface | Based on visual design add user interface components. | GWT widgets and panels and Root panel. |
| 4. Client side Management | Handling events of keyboard and mouse. | ClickHandler and KeyPressHandler interfaces. |
| 5. Code functionality (Client Side) | Creates and maintains one code for multiple browser implementations. | GWT Methods. |
| 6. Debugging | Debug the Java code before compiling it into JavaScript. | Development Mode. |
| 7. Apply style | Enhance the UI by applying CSS which changes style dynamically. Also apply some static element such as images etc. | GWT module, GWT themes, Style sheet, GWT methods, automatic resource inclusion. |
| 8. Compile and Deploy Application | Compile your Java code into JavaScript and test it in production mode. | GWT Compiler. |

**REFERENCES**

* 1. Java 2: The complete references – Herbert Schildt
  2. [www.javatpoint.com](http://www.javatpoint.com)
  3. [www.redhat.com](http://www.redhat.com)
  4. [www.datatrained.com](http://www.datatrained.com)
  5. [www.dotnetvishal.com](http://www.dotnetvishal.com)