|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
| **1** | **Annexure I– Micro Project Proposal** | **1-2** |
| 1.Aims/Benefits of the Micro-Project | 1 |
| 2. Course Outcome Addressed | 1 |
| 3.Proposed Methodology | 1 |
| 4. Action Plan | 2 |
| 5. Resources Required | 2 |
| 6. Name of Team Members with Roll No.’s | 2 |
| **2** | **Annexure II – Micro Project Report** | **3-8** |
| 1.Rationale | 3 |
| 2.Aims/Benefits of the Micro-Project | 3 |
| 3.Course Outcome Achieved | 3 |
| 4. Literature Review | 4 |
| 5.Actual Methodology Followed | 5 |
| 5.1 Flow chart | 5 |
| 5.2 Source code | 6-8 |
| 6.Actual  Resources Used | 8 |
| 7.Outputs of Micro-Projects | 9 |
| 8. Skill developed / Learning out of this Micro-Project | 10 |
| 9. Applications of this Micro-Project | 10 |

**Annexure I**

**Micro Project Proposal**

**“Java Programming”**

1. **Aims/Benefits of the Micro-Project:**

1) Gain an understanding of different types of operating systems and their functionalities.

2) Comprehend the requirements and design considerations for various types of operating systems.

3) Analyze the strengths and weaknesses of different operating system types in different contexts.

4) Apply the knowledge of operating systems to select an appropriate system for specific requirements or scenarios.

**2. Course Outcome Addressed:**

1. CO1- Develop program using GUI framework (AWT and Swing)
2. CO2- Handle events of AWT and Swing Components
3. CO3- Develop programs to handle events in Java Programming.
4. CO4- Develop Java Programs using Networking Concepts
5. CO5- Develop programs using Database.
6. CO6- Develop programs using Servlets.

**3. Proposed Methodology:**

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages.

**4. Action Plan**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Details of Activity** | **Planned**  **Start date** | **Planned**  **Finish date** | **Name of Responsible Team Members** |
| 1 | Search the topic | 29/08/2022  4:00pm-5:00pm | 05/09/2022  4:00pm-5:00pm |  |
| 2 | Search the information | 12/09/2022  4:00pm-5:00pm | 19/09/2022  4:00pm-5:00pm |  |
| 3 | Algorithm developing | 26/09/2022  4:00pm-5:00pm | 03/10/2022  4:00pm-5:00pm |  |
| 4 | Flowchart developing | 10/10/2022  4:00pm-5:00pm | 15/10/2022  4:00pm-5:00pm | Harsh Moreshwar Kale |
| 5 | Function making | 31/10/2022  4:00pm-5:00pm | 07/11/2022  4:00pm-5:00pm |  |
| 6 | Coding developing | 14/11/2022  4:00pm-5:00pm | 21/11/2022  4:00pm-5:00pm |  |
| 7 | Debugging | 28/11/2022  4:00pm-5:00pm | 05/12/2022  4:00pm-5:00pm |  |
| 8 | Finalizing Project with its report | 12/12/2022  4:00pm-5:00pm | 19/12/2022  4:00pm-5:00pm |  |

**5. Resources Required:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Name of resource / material** | **Specification** | **Quantity** | **Remarks** |
| 1 | Computer | WINDOWS 11, 8GB RAM, 160GB HDD | 1 |  |
| 2 | Operating System | WINDOWS 11 | 1 |  |
| 3 | Compiler | Turbo C/GCC/VS code | 1 |  |
| 4 | Browser | Chrome | 1 |  |

**Names of Team Members with Roll No.’s:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.**  **No.** | **Enrollment No.** | **Name of Team Member** | **Roll No.** |
| 1 | 2110950051 | Harsh Moreshwar Kale | 3 |

**Mr. Sugare D. D.**

**Name and Signature of the Teacher**

**Annexure – II**

**Micro-Project Report**

**“Java Programming”**

1. **Rationale:**

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture.

**2. Aims/Benefits of the Micro-Project:**

1. Understanding the computer operating system mechanism.

2. To break down larger projects into smaller, manageable tasks or sub-projects.

3. To understand which algorithm is used in operating system to priories the process.

**3. Course Outcomes Achieved:**

1) CO1 - Understanding of job scheduling techniques and resource allocation mechanisms in batch systems.

2) CO2 – Understanding of time-sharing concepts, context switching, and handling user interactions.

3) CO3 – Understanding of real-time scheduling algorithms, task prioritization, and meeting deadlines.

4) CO4 – Familiarity with CPU scheduling algorithms and memory management in multi-programmed systems.

5) CO5 – Familiarity with inter-processor communication, synchronization, and load balancing techniques in multiprocessor systems.

**4. Literature Review:**

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

**5 Actual Methodology Followed:**

**5.1 Flow Chart:**



**5.2 Source Code:**

**6. Actual Resources Used:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Name of resource / material** | **Specification** | **Quantity** | **Remarks** |
| 1 | Computer | WINDOWS 11,8GB RAM, 160GB HDD | 1 |  |
| 2 | Operating System | WINDOWS 11 | 1 |  |
| 3 | Compiler | Turbo C/GCC/VS code | 1 |  |
| 4 | Browser | Chrome | 1 |  |

**7. Outputs of Micro-Projects:**

**8. Skill developed / Learning out of this Micro-Project:**

There are so many thing that we learn from this project of

1. We can understand the importance of code optimization for enhancing the performance of our Java applications..

2. We can comprehend the significance of implementing proper error handling mechanisms to ensure the robustness of our software.. .

3. We can appreciate the value of using design patterns to improve the maintainability and scalability of our Java projects..

4. We can acknowledge the benefits of using frameworks like Spring to simplify the development process and enhance the modularity of our code.

5. Java teach us to realize the importance of conducting thorough testing and debugging to ensure the reliability and stability of our Java programs.. .

**9. Applications of this Micro-Project:**

1. Advanced Java micro-projects can be used to build dynamic and interactive web applications.
2. This includes developing server-side components using Java Servlets, JavaServer Pages (JSP), JavaServer Faces (JSF), and Java Persistence API (JPA). You can create web applications for e-commerce, content management systems, social networking platforms, and more..
3. Advanced Java micro-projects can involve developing IoT solutions using Java and frameworks like Eclipse IoT or MQTT (Message Queuing Telemetry Transport)..

\*\*\*\*\*\*\*\*\*