Government of Pakistan

**National Vocational and Technical Training Commission**

**Prime Minister’s Youth Skills Development Program**

"Skills for All"



**Course Contents / Lesson Plan**

**Course Title:** Docker Certified Associate

**Duration:** 3 Month

|  |  |
| --- | --- |
| **Trainer Name** |  |
| **Author** | **Tahir Ali (Instuctor-Corvit)** |
| **Course Title** | **Docker Certified Associate** |
| Objectives and Expectations | * **Docker is a popular platform for developing, shipping, and running applications in containers. Containers provide a lightweight and portable way to package and isolate applications and their dependencies, allowing them to run consistently across different environments. The primary objectives and expectations of Docker include:** * **Containerization:** * **Isolation: Docker containers encapsulate an application and its dependencies, ensuring that it runs consistently across various environments.** * **Portability: Containers can run on any system that supports Docker, providing a consistent environment from development to production.** * **Efficiency:** * **Resource Optimization: Containers share the host OS kernel, which makes them more efficient in terms of resource utilization compared to virtual machines.** * **Quick Start-Up: Containers can start up in seconds, allowing for rapid deployment and scaling.** * **DevOps and Continuous Integration/Continuous Deployment (CI/CD):** * **Consistency: Docker ensures consistency between development, testing, and production environments, reducing the "it works on my machine" problem.** * **Automated Builds: Docker facilitates the creation of Docker images through Dockerfiles, allowing for automated and reproducible builds.** * **Microservices Architecture:** * **Modularity: Docker supports a microservices architecture by allowing developers to break down applications into smaller, independently deployable and scalable components.** * **Orchestration:** * **Docker Swarm and Kubernetes: Docker can be integrated with orchestration tools like Docker Swarm or Kubernetes to manage and scale containerized applications across a cluster of machines.** * **Version Control and Rollback:** * **Image Versioning: Docker images can be versioned, providing a mechanism for version control and rollback in case of issues with newer releases.** * **Security:** * **Isolation: Containers provide process isolation, reducing the attack surface and enhancing security.** * **Immutable Infrastructure: Docker promotes the concept of immutable infrastructure, where containers are treated as disposable, and changes are made by creating new containers rather than modifying existing ones.** * **Collaboration:** * **Registry Services: Docker Hub and other container registries allow developers to share and collaborate on pre-built container images.** * **Cost Reduction:** * **Resource Utilization: Docker's efficient resource utilization can lead to cost savings in terms of infrastructure requirements.**   **Ecosystem and Community:**  **Extensive Ecosystem: Docker has a vast ecosystem of tools and technologies that enhance its functionality.**  **Community Support: Docker has a large and active community that contributes to its development and provides support.**  **In summary, Docker aims to provide a consistent and efficient environment for developing, packaging, and running applications, promoting collaboration, scalability, and automation in the software development lifecycle.** |
| Entry-level of trainees | **Four-year degree (bachelor's degree or global equivalent)** |
| **Learning Outcomes of the course** | **The learning outcomes of a Docker course typically revolve around gaining a comprehensive understanding of Docker's key concepts, features, and practical skills for using Docker in various scenarios. Here are common learning outcomes you might expect from a Docker course:**  **Understanding Containerization:**  **Define what containers are and understand the advantages of containerization over traditional virtualization.**  **Docker Architecture:**  **Explain the architecture of Docker, including the Docker daemon, Docker client, Docker images, and Docker containers.**  **Installation and Setup:**  **Install Docker on different operating systems (Windows, Linux, macOS) and configure basic settings.**  **Docker Images:**  **Create Docker images using Dockerfiles and understand the layers within images.**  **Pull and push Docker images from/to Docker Hub or other container registries.**  **Container Lifecycle:**  **Start, stop, and restart containers.**  **Manage the lifecycle of Docker containers, including pausing, removing, and inspecting containers.**  **Networking in Docker:**  **Understand Docker networking concepts, including bridge networks, host networks, and overlay networks.**  **Configure networking for Docker containers.**  **Storage in Docker:**  **Manage data persistence using Docker volumes.**  **Understand storage drivers and their impact on container storage.**  **Docker Compose:**  **Define multi-container applications using Docker Compose.**  **Orchestrate multiple containers as a part of a single application.**  **Docker Swarm:**  **Understand the basics of Docker Swarm for orchestrating and scaling containers.**  **Deploy and manage services in a Docker Swarm.**  **Security Best Practices:**  **Implement security best practices for Docker, including container isolation, user privileges, and image vulnerability scanning.**  **Integration with CI/CD:**  **Integrate Docker into a Continuous Integration/Continuous Deployment (CI/CD) pipeline.**  **Automate the build and deployment processes using Docker.**  **Troubleshooting:**  **Identify and resolve common issues and errors related to Docker containers.**  **Use Docker CLI and logs for troubleshooting.**  **Scalability and Load Balancing:**  **Scale applications by distributing containers across multiple hosts.**  **Implement load balancing for containerized applications.**  **Monitoring and Logging:**  **Implement monitoring and logging solutions for Docker containers.**  **Utilize tools like Docker Stats, Prometheus, and ELK stack for container monitoring.**  **Community and Resources:**  **Know where to find additional resources, documentation, and community support for Docker-related topics.**  **By the end of a Docker course, participants should have gained practical hands-on experience and be well-equipped to use Docker in real-world scenarios, whether for development, testing, or production deployments.** |
| **­­­Course Execution Plan** | **The total duration of the course: 3 months (12 Weeks)**  **Class hours: 4 hours per day**  **Theory: 20%**  **Practical: 80%**  **Weekly hours: 20 hours per week**  **Total contact hours: 240 hours** |
| **Companies offering jobs in the respective trade** | **Here are some types of companies that frequently seek professionals with Docker expertise:**   1. **Technology and Software Companies:**    * **Amazon Web Services (AWS)**    * **Microsoft**    * **Google Cloud Platform (GCP)**    * **IBM**    * **Red Hat**    * **Docker Inc.** 2. **IT Consulting and Services:**    * **Accenture**    * **Capgemini**    * **Deloitte**    * **Cognizant**    * **Infosys**    * **Wipro** 3. **Financial Services:**    * **JPMorgan Chase**    * **Goldman Sachs**    * **Bank of America**    * **Morgan Stanley** 4. **Healthcare and Pharmaceuticals:**    * **Pfizer**    * **Johnson & Johnson**    * **Siemens Healthineers**    * **Cerner Corporation** 5. **E-commerce and Retail:**    * **Amazon**    * **Walmart**    * **eBay**    * **Shopify** 6. **Telecommunications:**    * **AT&T**    * **Verizon**    * **T-Mobile**    * **Vodafone** 7. **Transportation and Logistics:**    * **FedEx**    * **UPS**    * **DHL**    * **Maersk** 8. **Social Media and Technology Platforms:**    * **Facebook**    * **Twitter**    * **LinkedIn**    * **Spotify** 9. **Startups:**    * **Many startups in various industries adopt containerization and Docker for their development and deployment processes.** 10. **Government and Defense Contractors:**     * **Companies providing technology solutions to government agencies and defense organizations often seek Docker skills.**   **When searching for Docker-related jobs, you can use job search platforms like LinkedIn, Indeed, Glassdoor, and specialized IT job boards. Additionally, exploring company career pages and attending industry-related events or conferences can provide valuable insights into job opportunities.** |
| **Job Opportunities** | * Information Technology * Construction * Healthcare * Finance and Banking * Engineering * Manufacturing * Consulting * Telecommunications * Energy and Utilities: |
| **No of Students** | 25 |
| **Learning Place** | Classroom / Lab |
| **Instructional Resources** | 1. Docker Tutorials : 2. https://www.youtube.com/watch?v=RqTEHSBrYFw   **Online Course Website**   |  |  | | --- | --- | | **Udemy:** | Docker and Containers: | | **Coursera:** | Docker Mastery | | **Pluralsight:** | Docker Deep Dive | | **LinkedIn Learning:** | Learning Docker | | **edX:** | [edX.org](https://www.edx.org/)/ Introduction to Docker | | **Docker's Official Training** | Docker Training | | **Packt:** | Docker for Cross-Platform | | **Skillshare:** | Docker and Containers: The Complete Guide | |  |  | |

**MODULES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Scheduled Weeks** | **Module Title** | **Days** | **Hours** | **Learning Units** | **Home Assignment** |
| **Week 1** | Introduction to Docker | Day 1 | Hour 1 | Introduction to Containerization | * **Task 1**   *Details may be seen at Annexure-I* |
| Hour 2 | Understanding containerization concepts |
| Hour 3 | Advantages of containerization |
| Hour 4 | Revision/Questions |
| Day 2 | Hour 1 | Docker Architecture and Installation |
| Hour 2 | Overview of Docker architecture |
| Hour 3 | Installing Docker |
| Hour 4 | Revision/Questions |
| Day 3 | Hour 1 | Docker Images |
| Hour 2 | Working with Docker images |
| Hour 3 | Managing Docker containers |
| Hour 4 | Revision/Questions |
| Day 4 | Hour 1 | Docker Networking Basics |
| Hour 2 | Docker networking modes |
| Hour 3 | Managing Docker networks |
| Hour 4 | Revision/Questions |
| Day 5 | Hour 1 | Installing Docker on various platforms |
| Hour 2 | Docker Images and Containers |
| Hour 3 | Introduction Docker CLI |
| Hour 4 | Revision/Questions |
| **Week 2** | **Working with Docker** | Day 1 | Hour 1 | Docker CLI Fundamentals | * **Task 2**   *Details may be seen at Annexure-I* |
| Hour 2 | Essential Docker CLI commands |
| Hour 3 | Managing Docker containers from the command line |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Docker Volumes and Storage |
| Hour 2 | Understanding Docker storage drivers |
| Hour 3 | Working with Docker volumes |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Dockerfile Basics |
| Hour 2 | Introduction to Docker file |
| Hour 3 | Writing Docker files |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Docker Compose Fundamentals |
| Hour 2 | Introduction to Docker Compose |
| Hour 3 | Writing Docker Compose YAML files |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | Practical |
| **Week 3** | **Docker Orchestration** | Day 1 | Hour 1 | Introduction to Docker Swarm | * **Task 3**   *Details may be seen at Annexure-I* |
| Hour 2 | Overview of Docker Swarm |
| Hour 3 | Creating a Docker Swarm  cluster |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Deploying Applications with Swarm |
| Hour 2 | Deploying services to Docker Swarm |
| Hour 3 | Service scaling and updates in Swarm |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Introduction to Kubernetes |
| Hour 2 | Overview of Kubernetes architecture |
| Hour 3 | Deploying Kubernetes clusters |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Deploying Applications with Kubernetes |
| Hour 2 | Creating Kubernetes deployments |
| Hour 3 | Service discovery and load balancing in Kubernetes |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | Practical Revision |
| **Week 4** | **Docker Security and Monitoring** | Day 1 | Hour 1 | Docker Security Principles | * **Task 4**   *Details may be seen at Annexure-I* |
| Hour 2 | Overview of Docker security features |
| Hour 3 | Best practices for securing Docker environments |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Container Monitoring and Logging |
| Hour 2 | Monitoring Docker containers and hosts |
| Hour 3 | Configuring logging for Docker containers |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Docker Swarm Security and Monitoring |
| Hour 2 | Securing Docker Swarm clusters |
| Hour 3 | Monitoring Docker Swarm services and nodes |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Kubernetes Security and Monitoring |
| Hour 2 | Securing Kubernetes clusters |
| Hour 3 | Monitoring Kubernetes pods, services, and nodes |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | LAB Practical |
| **Week 5** | **Advanced Docker Concepts** | Day 1 | Hour 1 | Docker Networking Advanced | * **Task 5**   *Details may be seen at Annexure-I* |
| Hour 2 | Advanced Docker networking concepts |
| Hour 3 | Docker overlay networks |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Docker Storage Management |
| Hour 2 | Advanced Docker storage management |
| Hour 3 | Docker volume plugins |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Docker High Availability |
| Hour 2 | High availability strategies for Docker |
| Hour 3 | Docker Swarm and Kubernetes HA configurations |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Docker Performance Tuning |
| Hour 2 | Performance optimization techniques for Docker |
| Hour 3 | Monitoring and optimizing Docker resource usage |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | LAB Practical |
| **Week 6** | **Continuous Integration and Deployment** | Day 1 | Hour 1 | Introduction to CI/CD with Docker | * **Task 6**   *Details may be seen at Annexure-I* |
| Hour 2 | Overview of CI/CD concepts |
| Hour 3 | Integrating Docker with CI/CD pipelines |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Building Docker Images in CI/CD |
| Hour 2 | Automated Docker image builds |
| Hour 3 | Docker image versioning and tagging strategies |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Docker Deployment Strategies |
| Hour 2 | Blue-green deployment with Docker |
| Hour 3 | Canary deployment with Docker |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Docker Monitoring Tools |
| Hour 2 | Overview of Docker monitoring tools |
| Hour 3 | Setting up monitoring and alerting for Docker containers |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | Practical / Lab |
| **Week 7** | **Docker Ecosystem and Integrations** | Day 1 | Hour 1 | Docker Compose in Production | * **Task 7**   *Details may be seen at Annexure-I* |
| Hour 2 | Using Docker Compose for production deployments |
| Hour 3 | Best practices for Docker Compose usage |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Docker Orchestration with Ansible |
| Hour 2 | Automating Docker Swarm and Kubernetes deployments with Ansible |
| Hour 3 | Managing Docker containers with Ansible |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Docker with Terraform |
| Hour 2 | Managing Docker infrastructure with Terraform |
| Hour 3 | Provisioning Docker clusters using Terraform |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Docker in Cloud Environments |
| Hour 2 | Deploying Docker containers on AWS, Azure, and Google Cloud Platform |
| Hour 3 | Integrating Docker with cloud-native services |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | Practical / Lab |
| **Week 8** | **DOS/DDOS & SQL Injection Attack** | Day 1 | Hour 1 | Project Work: Designing Dockerized Application | * **Task 8**   *Details may be seen at Annexure-I* |
| Hour 2 | Designing a containerized application architecture |
| Hour 3 | Selecting appropriate Docker tools and services |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Project Work: Implementing Dockerized Application |
| Hour 2 | Implementing Docker best practices in the application |
| Hour 3 | Testing and debugging the Dockerized application |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Project Work: Deployment and Optimization |
| Hour 2 | Deploying the Dockerized application to production |
| Hour 3 | Performance optimization and scaling strategies |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Review and Feedback |
| Hour 2 | Reviewing key Docker concepts and topics covered |
| Hour 3 | Providing feedback on the course content and delivery |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-2 | Practical / Lab |
| **Week 9** | **Docker Networking and Security** | Day 1 | Hour 1 | Advanced Docker Networking | * **Task 9**   *Details may be seen at Annexure-I* |
| Hour 2 | Docker overlay networks and multi-host networking |
| Hour 3 | Implementing network segmentation and security in Docker |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Docker Security Best Practices |
| Hour 2 | Hardening Docker containers and hosts |
| Hour 3 | Implementing security policies and controls |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Docker Secrets Management |
| Hour 2 | Managing sensitive data in Docker applications |
| Hour 3 | Docker secrets and configuration management |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Docker Security Scanning |
| Hour 2 | Scanning Docker images for vulnerabilities |
| Hour 3 | Implementing continuous security scanning in Docker environments |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | Practical / Lab |
| **Week 10** | **Docker Performance Optimization** | Day 1 | Hour 1 | Docker Resource Management | * **Task 10**   *Details may be seen at Annexure-I* |
| Hour 2 | Docker resource constraints and limitations |
| Hour 3 | Managing CPU, memory, and I/O resources in Docker |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Docker Performance Tuning |
| Hour 2 | Identifying and resolving performance bottlenecks in Docker |
| Hour 3 | Optimizing Docker container and image size |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Docker Load Balancing |
| Hour 2 | Load balancing strategies for Docker containers |
| Hour 3 | Implementing Docker container orchestration with load balancers |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Docker Logging and Monitoring |
| Hour 2 | Logging best practices for Docker containers |
| Hour 3 | Implementing monitoring solutions for Docker environments |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | Practical / Lab |
| **Week 11** | **Docker Advanced Concepts** | Day 1 | Hour 1 | Docker Storage Drivers | * **Task 11**   *Details may be seen at Annexure-I* |
| Hour 2 | Understanding Docker storage drivers and options |
| Hour 3 | Selecting appropriate storage drivers for different use cases |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Docker Networking Plugins |
| Hour 2 | Extending Docker networking with plugins |
| Hour 3 | Implementing custom networking solutions in Docker |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Docker Plugin Development |
| Hour 2 | Developing custom plugins for Docker |
| Hour 3 | Integrating third-party services with Docker using plugins |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Docker Community and Ecosystem |
| Hour 2 | Exploring the Docker community and ecosystem |
| Hour 3 | Contributing to Docker projects and initiatives |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | Practical / Lab |
| **Week 12**  **Week 13** | **Docker Project Work and Review** | Day 1 | Hour 1 | Project Work: Designing Dockerized Application | * **Task 12**   *Details may be seen at Annexure-I*  **Final Project** |
| Hour 2 | Designing a containerized application architecture |
| Hour 3 | Selecting appropriate Docker tools and services |
| Hour 4 | Revision/Questions/Practice |
| Day 2 | Hour 1 | Project Work: Implementing Dockerized Application |
| Hour 2 | Implementing Docker best practices in the application |
| Hour 3 | Testing and debugging the Dockerized application |
| Hour 4 | Revision/Questions/Practice |
| Day 3 | Hour 1 | Project Work: Deployment and Optimization |
| Hour 2 | Deploying the Dockerized application to production |
| Hour 3 | Performance optimization and scaling strategies |
| Hour 4 | Revision/Questions/Practice |
| Day 4 | Hour 1 | Final Review and Feedback |
| Hour 2 | Reviewing key Docker concepts and topics covered |
| Hour 3 | Providing feedback on the course content and delivery |
| Hour 4 | Revision/Questions/Practice |
| Day 5 | Hour 1-4 | Practical / Lab |
|  | **Final Exam and Assessment** |

| **Task No.** | **Task** | **Description** | **Week** |
| --- | --- | --- | --- |
|  | **Docker Installation** | * Install Docker on a Linux server following best practices. | **Week 1** |
|  | **Building Docker Image** | * Create a Dockerfile to build a custom image for a Node.js application. | **Week 2** |
|  | **Docker Networking** | * Configure a custom Docker network for communication between containers. | **Week 3** |
|  | **Container Orchestration** | * Deploy a multi-container application using Docker Compose. | **Week 4** |
|  | **Swarm Cluster Setup** | * Set up a Docker Swarm cluster with at least three nodes. | **Week 5** |
|  | **Swarm Service Deployment** | * Deploy a service on Docker Swarm, ensuring high availability. | **Week 6** |
|  | **Docker Security Scanning** | * Perform a security scan on a Docker image using Docker Security Scan. | **Week 7** |
|  | **Docker Volumes** | * Use Docker volumes to persist data between container restarts. | **Week 8** |
|  | **Kubernetes Basics** | * Deploy a simple application on a Kubernetes cluster. | **Week 9** |
|  | **Kubernetes Pods** | * Create and manage pods in a Kubernetes cluster. | **Week10** |
|  | **Kubernetes Deployments** | * Implement rolling updates for a Kubernetes Deployment. | **Week11** |
|  | **Kubernetes ConfigMaps and Secrets** | * Utilize ConfigMaps and Secrets to manage application configuration. | **Week12** |
| **13.** | **Docker CI/CD Integration** | * Set up a CI/CD pipeline using Docker to automate image builds and deployments. |  |
| **14.** | **Monitoring Docker Containers** | * Implement monitoring for Docker containers using Prometheus and Grafana. |  |
| **15.** | **Troubleshooting Docker Containers** | * Troubleshoot and resolve common issues with Docker containers and services. |  |
| **16** | **Final Exam** | * Create a Dockerfile for a Python application that installs dependencies, sets environment variables, copies the application code, and exposes the necessary port. |  |

**Annexure-IV:**

**Workplace/Institute Ethics Guide**

Work ethic is a standard of conduct and values for job performance. The modern definition of what constitutes good work ethics often varies.  Different businesses have different expectations. Work ethic is a belief that hard work and diligence have a moral benefit and an inherent ability, virtue, or value to strengthen character and individual abilities. It is a set of values-centered on the importance of work and manifested by determination or desire to work hard.

The following ten work ethics are defined as essential for student success:

1. **Attendance:**Be at work every day possible, plan your absences don’t abuse leave time. Be punctual every day.
2. **Character:**Honesty is the single most important factor having a direct bearing on the final success of an individual, corporation, or product. Complete assigned tasks correctly and promptly. Look to improve your skills.
3. **Team Work:**

The ability to get along with others including those you don’t necessarily like. The ability to carry your weight and help others who are struggling. Recognize when to speak up with an idea and when to compromise by blend ideas together.

1. **Appearance:**Dress for success set your best foot forward, personal hygiene, good manner, remember that the first impression of who you are can last a lifetime
2. **Attitude:**Listen to suggestions and be positive, accept responsibility. If you make a mistake, admit it. Values workplace safety rules and precautions for personal and co-worker safety. Avoids unnecessary risks. Willing to learn new processes, systems, and procedures in light of changing responsibilities.
3. **Productivity:**Do the work correctly, quality and timelines are prized. Get along with fellows, cooperation is the key to productivity. Help out whenever asked, do extra without being asked. Take pride in your work, do things the best you know-how. Eagerly focuses energy on accomplishing tasks, also referred to as demonstrating ownership. Takes pride in work.
4. **Organizational Skills:**

Make an effort to improve, learn ways to better yourself. Time management; utilize time and resources to get the most out of both. Take an appropriate approach to social interactions at work. Maintains focus on work responsibilities.

1. **Communication:**Written communication, being able to correctly write reports and memos.  
   Verbal communications,being able to communicate one on one or to a group.
2. **Cooperation:**Follow institute rules and regulations, learn and follow expectations. Get along with fellows, cooperation is the key to productivity. Able to welcome and adapt to changing work situations and the application of new or different skills.
3. **Respect:**Work hard, work to the best of your ability. Carry out orders, do what’s asked the first time. Show respect, accept, and acknowledge an individual’s talents and knowledge. Respects diversity in the workplace, including showing due respect for different perspectives, opinions, and suggestions.