INDIRA GANDHI NATIONAL OPEN UNIVERSITY

Regional Centre Delhi-1

Format for Assignment Submission

For Term End Exam June / December- December__ (Year) _2025___

(Please read the instructions given below carefully before submitting assignments)

1. Name of the Student : Mark L F Tlau
2. Enrollment Number : 2550670762
3. Programme Code : BCA_NEW
4. Course Code : BCS111
   (Use this format course-wise separately)
5. Study Centre Code : 07162P
6. Name of the Study Centre With complete address : Mohyal Education and Research Institute, Qutab Institutional Area, New Delhi
7. Mobile Number : +916009341754
8. E-mail ID : wintersunset95@gmail.com
9. Details if this same assignment has been submitted anywhere else also :
10. Above information is cross checked and it is correct: Yes/No (Yes)

Date of Submission: (Signature of the student)

A. Important Instructions:-

1. Please do not send any assignment at any email of the Regional Centre, it will not be considered.
2. Please avoid duplicacy. Do not re-submit the same assignment anywhere else or by any other means.
3. About the mode of submission of assignments, pl wait for instructions from IGNOU Hqtrs. As soon as we shall come to know, we will share it with all.
4. Please do not use plastic covers. Use plain A4 size pages for assignments for uniformity and better management with this cover page format on each assignment.
5. Please write your name and enrollment no. at the bottom of each page of your assignment.
6. Please retain a photocopy set of assignment submitted with you for record (may be asked to submit at later stage) and also keep the assignment submission receipt in safe custody.
7. If assignment awards are not updated in your Grade Card **within next 09 months**, please write to us at rcdelhi1@ignou.ac.in giving your complete details and attaching the proof of assignment submission.
8. Assignment Question Paper can be downloaded from: https://webservices.ignou.ac.in/assignments/

B. Compulsory sequence of the Assignment Set:

1. Duly Filled in Assignment Submission Cover Page (This Format Page).
2. Copy of IGNOU Identity Card.
3. Printout of valid/applicable assignment question paper.
4. Handwritten Assignment, written on both the sides of page (preferably plain A4 size).

Q1:

1. **Trace the evolution of computers from mechanical calculators to the fifth generation, and identify how each technological shift enabled a new kind of realworld application.**

Answer: The evolution of computers is a story of technological shifts, each enabling new real-world applications. The **Mechanical Era (1600s-1900s)**, based on gears and levers, first enabled the automation of calculation for mathematical tables. The **First Generation (1940s-1950s)** shifted to *vacuum tubes*, allowing for massive, complex electronic calculations, which were applied to military ballistics (ENIAC) and scientific research. The move to smaller, more reliable *transistors* marked the **Second Generation (1950s-1960s)**, reducing cost and enabling the first commercial business applications like payroll and inventory management. **Third Generation (1960s-1970s)** computers were built on the *Integrated Circuit (IC)*, which led to minicomputers (like the DEC PDP-8) and brought computing to smaller businesses and university labs. The **Fourth Generation (1970s-Present)** began with the *microprocessor*, putting an entire CPU on one chip. This shift democratized computing, enabling the personal computer (PC) and its applications like word processors, spreadsheets, and the internet. Finally, the **Fifth Generation (Present & Future)** is defined by *Parallel Processing and AI*, enabling complex modern applications like big data analysis, machine learning, and advanced scientific simulation.

2. **Draw a labelled diagram of the internal data flow in a digital computer and explain how instructions are fetched and executed using a practical example, such as calculating total marks.**

Answer: A digital computer's internal data flow, based on the Von Neumann architecture, involves a Central Processing Unit (CPU), Memory (RAM), and Input/Output (I/O) devices, all connected by buses (Address, Data, Control). The CPU processes instructions using the **Fetch-Decode-Execute Cycle**.

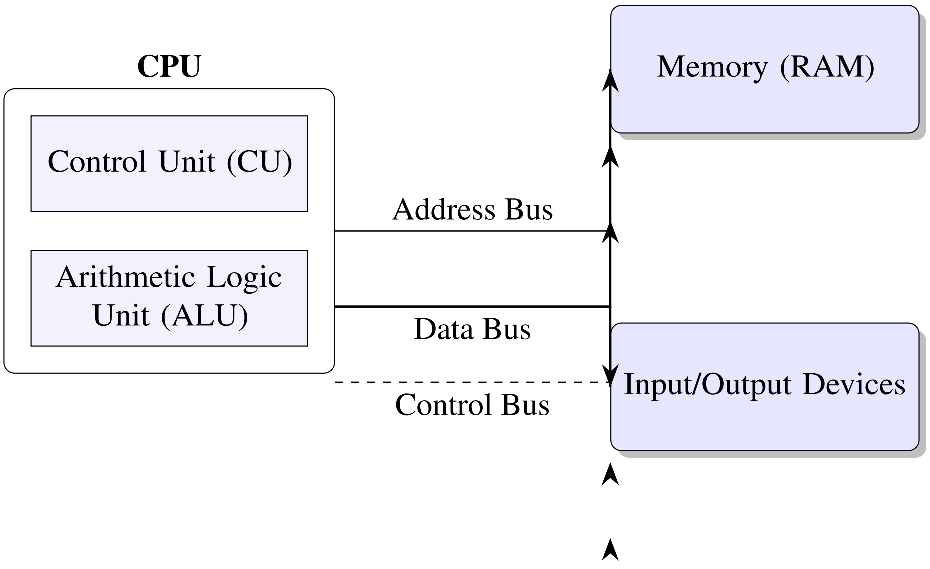

Figure 1: Internal Data Flow (Von Neumann Architecture)

Using the example 'Total = Marks1 + Marks2', the cycle is as follows:

(a) **Fetch:** The Control Unit (CU) requests the first instruction (e.g., 'LOAD Marks1') from its address in RAM. The instruction travels along the Data Bus to the CPU.

(b) **Decode:** The CU decodes the 'LOAD' instruction, understanding it needs to fetch data from the address of 'Marks1'.

(c) **Execute (Fetch Data):** The CU sends the 'Marks1' address on the Address Bus. The data (e.g., '80') travels back from RAM on the Data Bus to a CPU register (the Accumulator).

(d) **Fetch/Decode:** The CU fetches and decodes the next instruction, 'ADD Marks2'.

(e) **Execute (Add):** The CU fetches the data for 'Marks2' (e.g., '90'). The Arithmetic Logic Unit (ALU) adds this '90' to the '80' in the Accumulator, resulting in '170'.

(f) **Fetch/Decode/Execute (Store):** The final instruction, ‘STORE Total‘, is fetched and decoded. The CU executes it by sending the address for ‘Total‘ and the value ‘170‘ (from the Accumulator) back to RAM.

3. **Evaluate the impact of memory hierarchy (cache, RAM, secondary storage) on the performance of commonly used applications such as video editing or gaming.**

Answer: The memory hierarchy is essential for performance, acting as a bridge between the fast CPU and slow storage. In demanding applications like gaming or video editing, each level has a distinct impact. **Cache (L1/L2/L3)** is the fastest memory, and its impact is on real-time responsiveness. In gaming, it stores frequently used instructions (like game physics), preventing "stutter" and enabling a smooth frame rate. **RAM (Main Memory)** is the medium-speed "working area." Its impact is on multitasking and handling large active files. In video editing, the OS, the software, and the active video clips must fit in RAM. If RAM is insufficient (e.g., 8GB for a 4K project), the system must constantly swap data with the disk, causing severe slowdowns. Finally, **Secondary Storage (SSD/HDD)** is the slowest level, and its impact is on loading times. In gaming, an SSD will dramatically reduce initial game load screens by transferring large assets (textures, maps) to RAM much faster than an old HDD.

4. **Imagine you are designing a kiosk for railway ticket booking. Justify your selection of input and output devices based on user interaction, reliability, and efficiency.**

Answer: For a public railway kiosk, **input devices** must be robust and intuitive. The primary interface would be a **resistive touch screen**; it is chosen for high **reliability** against public use and vandalism, as it responds to pressure (unlike a capacitive screen) and works with any object. For payment, a **stainless steel PIN pad** is justified for its high **reliability** and security in entering sensitive PINs. Lastly, a **QR/Barcode Scanner** would be included for **efficiency**, allowing users to scan IDs, mobile payment codes, or existing tickets far more quickly and accurately than manual typing.

The **output devices** must provide clear, fast feedback. The main **display**, integrated with the touch screen, is the primary output, providing immediate visual feedback for good **user interaction**. A **thermal printer** is the ideal choice for printing tickets. Its justification is based on **efficiency** (it’s extremely fast and requires no ink/toner) and **reliability** (few moving parts), making it perfect for a high-volume kiosk. Finally, a **speaker** would be included to provide audio prompts, improving overall **user interaction** and accessibility.

5. **You have Rs. 35,000 to assemble a personal computer for online learning and basic office work. List the hardware components you would choose and justify your selection based on performance and cost.**

Answer: This budget is for the PC tower only, aiming for a responsive experience for multitasking (video calls, browser tabs, Office apps). The item format is best for this answer.

(a) **CPU: AMD Ryzen 5 5600G** (approx. Rs. 11,000) *Justification:* This CPU is the core of the build. Its integrated Vega 7 graphics are powerful enough for smooth 1080p video calls and office tasks, completely saving the **cost** of a separate graphics card. Its 6 cores provide excellent **performance** for multitasking.

(b) **Motherboard: B450M or A520M** (approx. Rs. 5,500) *Justification:* A **cost**-effective motherboard that is fully compatible with the 5600G. It provides all necessary features, including slots for NVMe SSD and 2-4 RAM sticks.

(c) **RAM: 16GB (2x8GB) DDR4 3200MHz** (approx. Rs. 4,500) *Justification:* 16GB is the sweet spot for multitasking **performance**. 8GB would be a bottleneck. Using two sticks (2x8GB) enables "dual-channel" mode, which significantly boosts the **performance** of the 5600G’s integrated graphics.

(d) **Storage: 1TB NVMe Gen3 SSD** (approx. Rs. 6,500) *Justification:* This is the single biggest factor for "fast-feeling" **performance**. An NVMe SSD makes boot times, opening applications (Word, Chrome), and finding files near-instantaneous. 1TB offers ample space at a great **cost**-per-GB.

(e) **Power Supply (PSU): 450W 80+ Bronze** (approx. Rs. 3,500) *Justification:* A reliable PSU from a brand like Cooler Master or Deepcool is critical for system stability. 450W is more than enough power for this efficient build, and the 80+ Bronze rating ensures good **cost**-efficiency.

(f) **Case: Basic ATX/mATX Mid-Tower** (approx. Rs. 4,000) *Justification:* A **cost**-effective case with decent airflow to house the components.

Total: approx. Rs. 35,000. This build prioritizes a fast SSD and 16GB of RAM, which will provide the best day-to-day performance for learning and office work.

Q2:

1. **Compare proprietary and open-source software models by analyzing two software products (e.g., Microsoft Office vs. LibreOffice) in terms of cost, accessibility, updates, and support.**

Answer: A comparison between the proprietary **Microsoft Office** and the open-source **LibreOffice** highlights the fundamental differences in their models. The most evident factor is **cost**: Microsoft Office requires a recurring subscription or a high one-time license fee, whereas LibreOffice is entirely free. This is linked to **accessibility**; Microsoft's source code is a protected trade secret, while LibreOffice's source code is open for anyone to view, modify, and distribute. This core difference dictates the approach to **updates and support**. Microsoft provides official, paid customer support and centrally controls all updates, pushing features as it sees fit. LibreOffice, conversely, relies on free, community-based support through public forums and wikis, with updates being developed collaboratively by a global volunteer community.

2. **Assume your system has both Linux and Windows installed. Discuss three scenarios where Linux offers more control or flexibility compared to Windows, and explain why.**

Answer: Linux provides significantly more granular control and flexibility than Windows in many scenarios. A primary example is **system updates**. Windows often forces mandatory updates and subsequent reboots at inconvenient times. In Linux, the user has 100% control: they can decide precisely *when* to update, *what* specific packages to update (e.g., update only the browser but not the kernel), and reboots are almost never required for updates. This control extends to **software and environment management**. On Windows, installing a development environment requires running multiple, separate '.exe' installers. On Linux, a package manager (like 'apt' or 'dnf') provides a central, scriptable interface to install and manage all software, giving superior control. Lastly, Linux offers greater flexibility in **system recovery**. If a Windows installation becomes corrupt, a user can boot into their dual-boot Linux partition, mount the Windows (NTFS) drive, and have full, unrestricted access to back up all their files or even edit Windows configuration files to attempt a repair.

3. **Reflect on how compilers and interpreters differ in handling programming errors. Provide an example where using an interpreter would be more beneficial during development.**

Answer: Compilers and interpreters differ fundamentally in error handling. A **compiler** (like for C++) translates the *entire* program into machine code before running it. During this "compile-time" phase, it scans all files and generates a complete list of every syntax or type-checking error. The program cannot run at all until every single error is fixed. In contrast, an **interpreter** (like for Python) translates and executes the program *line-by-line* at "run-time." It starts immediately, and execution only stops when it encounters a line with an error, reporting just that single error. This line-by-line approach is highly beneficial for rapid development. For example, if you are writing a data analysis script, you can run it, wait 30 seconds for it to load a large file, and then have it fail on a minor syntax error in your charting code. With an interpreter, you can fix that single line and immediately re-run, providing a much faster feedback loop than a compiler, which would force you to re-compile the entire application for every small change.

4. **Create a scenario (e.g., preparing an event budget) where both a word processor and spreadsheet are required. Describe how you would use each to accomplish the task effectively.**

Answer: A perfect scenario is preparing an official "Annual College Fest Budget Proposal" for the principal. Both tools are required for an effective workflow. I would first use the **spreadsheet (e.g., LibreOffice Calc)** to handle all financial calculations. Here, I would create a detailed table with columns for 'Item', 'Quantity', 'Cost per Unit', and 'Total Cost', using formulas to auto-calculate the total for each item and a 'SUM()'

function for the "Grand Total." This ensures all mathematics are 100% accurate and easily adjustable. After finalizing the numbers, I would use the **word processor (e.g., LibreOffice Writer)** to create the formal proposal document. This file would contain the formatted cover page, the event introduction, and the text justifying the budget. Finally, I would copy the completed budget table from the spreadsheet and paste it directly into the word processor. This combines the narrative and formatting strengths of the word processor with the calculation power of the spreadsheet.

5. **Design a simple database schema to store student attendance in a classroom. Mention tables, fields, and types. Also, mention one project management tool and how it can help in managing a classroom project.**

Answer: A simple, normalized database schema would require two tables to avoid redundant data.

Table 1: ‘Students‘ (Stores one record per student)

- ‘student_id‘ (Type: ‘INT‘, Primary Key)
- ‘first_name‘ (Type: ‘VARCHAR(100)‘)
- ‘last_name‘ (Type: ‘VARCHAR(100)‘)
- ‘roll_number‘ (Type: ‘VARCHAR(20)‘, Unique)

Table 2: ‘AttendanceLog‘ (Stores one record per student, per day)

- ‘log_id‘ (Type: ‘INT‘, Primary Key)
- ‘student_id_fk‘ (Type: ‘INT‘, Foreign Key references ‘Students.student_id‘)
- ‘attendance_date‘ (Type: ‘DATE‘)
- ‘status‘ (Type: ‘ENUM(’Present’, ’Absent’, ’Late’)‘)

For a project management tool, a team could use a Kanban board like **Trello**. To manage a classroom project, the group would create a Trello board with lists such as "To-Do," "In-Progress," and "Done." Each project task (e.g., "Research Topic," "Create PowerPoint Slides," "Write Script") becomes a "Card." Team members can be assigned to cards, add checklists, and set due dates. As a member begins a task, they drag their card to "In-Progress," and then to "Done" upon completion. This provides a simple, visual way for everyone, including the teacher, to track the project’s progress and see who is responsible for each part.

Q3:

1. **You have to set up a small office network with internet connectivity. Describe the networking devices (router, switch, etc.) and topology you would choose and justify your decision.**

Answer: For a small office network, the best choice is a **Star Topology**, which is reliable and easy to manage. The central device would be a **Router**. I would choose an integrated "small office/home office" (SOHO) router, which combines a router, a switch, and a wireless access point into one cost-effective device.

This single router serves three functions: 1. **Router:** It connects our entire office (the LAN) to the internet (the WAN) via the ISP’s modem. It manages all traffic and acts as the DHCP server, automatically assigning local IP addresses to all devices. 2. **Switch:** Its built-in Ethernet ports (typically 4-8) will form the core of the star topology. All wired devices, such as desktop PCs and network printers, will connect via cable directly to these ports. If more ports are needed, a secondary, standalone **Switch** can be plugged into one of these ports to expand the network. 3. **Wireless Access Point (WAP):** The built-in Wi-Fi provides internet connectivity for mobile devices like laptops, tablets, and phones.

This star topology is justified because it is robust; if one device’s cable fails, only that device loses connection, and the rest of the network remains operational. It’s also efficient and easy to troubleshoot.

2. **Suppose you are tasked with designing a personal learning portal. Identify three internet services you would integrate (e.g., cloud storage, video conferencing, email notifications) and justify their**

use.

Answer: A personal learning portal should centralize all aspects of a student's workflow. Three essential internet services to integrate would be cloud storage, an authentication service, and a real-time communication API.

First, I would integrate a **Cloud Storage** service like Google Drive or Dropbox. This is justified because it allows the student to store and access all their course materials—notes, e-books, and assignment drafts—from any device. It also provides version history and a reliable backup, preventing catastrophic data loss from a local hard drive failure.

Second, I would use a social **Authentication Service** like "Sign in with Google" or "Sign in with GitHub" (OAuth). This is justified because it enhances both security and convenience. The user doesn't need to create and remember yet another password, and the portal doesn't need to securely store user credentials, offloading that complex security burden to a trusted provider.

Third, I would integrate a **Real-Time Communication** service (like WebRTC or a third-party API) for video conferencing. This is justified as it transforms the portal from a static file repository into an interactive learning hub. It enables live one-on-one sessions with tutors, participation in virtual study groups, and viewing live-streamed lectures, all from within the same interface.