|  |  |
| --- | --- |
| **SESSION** | **AUGUST 2023** |
| **PROGRAM** | **MASTER OF COMPUTER APPLICATIONS (MCA)** |
| **SEMESTER** | **I** |
| **COURSE CODE & NAME** | **DCA6102 – PROGRAMMING IN C** |
| **CREDITS** | **4** |
| **NUMBER OF ASSIGNMENTS & MARKS** | **02****30 Marks each** |

**1.(a) Explain different generations of computer system.**

**(b) What are the differences between concurrent and parallel execution?**

**Ans 1.**

**(a) Different Generations of Computer Systems:**

The history of computers can be divided into several generations, each marked by significant technological advancements and changes in hardware and software architecture. These generations are typically categorized as follows:

1. First Generation (1940s-1950s):

Vacuum Tubes: The earliest Its Half solved only

Buy Complete from our online store

<https://smuassignment.in/online-store/>

MUJ Fully solved assignment available for**session SEPT 2023.**

Lowest price guarantee with quality.

Charges**INR 198 only per assignment.**For more information you can get via mail or Whats app also

Mail id is aapkieducation@gmail.com

Our website www.smuassignment.in

After mail, we will reply you instant or maximum

1 hour.

Otherwise you can also contact on our

whatsapp no 8791490301.

**2 (a) Explain zero, one, two and three address instruction with the help of given instruction**

**X=(A/B)+(C\*D)**

**(b) Discuss different types of addressing modes in detail.**

**Ans:**

**2 (a) Zero, One, Two, and Three Address Instructions:**

In computer architecture and assembly language programming, instructions are used to perform various operations on data. The number of addresses or operands an instruction uses can vary, and this categorization is often referred to as zero, one, two, or three-address instructions. Let's use the given expression, X = (A/B) + (C\*D), to illustrate these types of instructions:

1. Zero-Address Instructions:

 **3 (a) Explain different types of techniques to handle hazards in pipelining.**

**(b) Differentiate between unconditional and conditional branch.**

**Ans 3.**

**3 (a) Techniques to Handle Hazards in Pipelining:**

Pipelining is a technique used in modern microprocessors to improve instruction execution throughput by breaking down the instruction execution process into stages. However, pipelining can introduce hazards, which are situations where the next instruction cannot proceed smoothly due to dependencies or conflicts with previous instructions. To handle hazards in pipelining, several

**4. Describe various features of the C programming language.**

**Ans 4.**

Various Features of the C Programming Language:

C is a powerful and versatile programming language that has been widely used for several decades. It is known for its simplicity, efficiency, and portability. Here are various features of the C programming language:

1. **Procedural Language**: C is a procedural programming language, which means it is organized

**5. Explain various flow control statements in C with examples.**

**Ans 5.**

**Various Flow Control Statements in C with Examples:**

Flow control statements in C are used to control the execution order of instructions within a program. Here are some commonly used flow control statements with examples:

**1. if Statement**:

**6. Define a function. List and explain the categories of user-defined functions.**

**7. Define an array. How to initialize a one-dimensional array? Explain with suitable examples.**

Definition of a Function and Categories of User-Defined Functions:

A function in programming is a self-contained block of code that performs a specific task or set of tasks. It is a reusable unit of code that can be called from various parts of a program. Functions enhance code modularity, readability, and reusability. In C and many other programming languages, functions are defined using a set of rules and syntax. There are several categories of user-defined functions:

1. **Function Declaration/Prototype**:
	* A function declaration, also known as a function prototype, specifies the function's name, return type, and the types of its parameters. It tells the compiler about the function's