

HBTU	Harcourt Butler Technical University, Kanpur		MID SEM I
Branch	CE/IT/CS/EE/ET/ME	Program	B.Tech.
Course Name	Introduction to Computer Science	Semester	I
Course Code	NCS 101	Year/Session	2025–26
Time: 1 hr		Max. Marks	15
Knowledge Level (KL)	K1: Remembering K2: Understanding	K3: Applying K4: Analysing	K5: Evaluating K6: Creating

Note: Attempt all questions.

Q. No	Questions	Marks	COs	KL
1	Solve the following: (i) $(219.5)_{10} = ()_2$ (ii) $(A12B)_{16} = ()_8$ (iii) $(1100110101)_{\text{Gray code}} = ()_2$	3	CO3	K6
2	Differentiate between System Software and Utility Software with an example.	3	CO2	K1 K2
3	Explain how microcomputers differ from minicomputers in terms of architecture and user handling.	3	CO3	K1 K2
4	Construct the truth table for the Boolean expression $Y = A \cdot (B + C)$.	3	CO3	K6
5	Differentiate between Assembler, Compiler, and Interpreter.	3	CO2	K1

Introduction to Computer Science: Mid-sem 1, 2025–26

Model answers prepared for study; not an official university answer key.

1. Number systems

(i) $219 = 128 + 64 + 16 + 8 + 2 + 1$, and $0.5 = 2^{-1}$. Hence $(219.5)_{10} = (11011011.1)_2$.

(ii) Convert hexadecimal digits to four-bit groups, then regroup in threes from the right:

$$A12B_{16} = 1010\ 0001\ 0010\ 1011_2 = 001\ 010\ 000\ 100\ 101\ 011_2 = 120453_8.$$

(iii) For Gray to binary, copy the first bit and obtain each later binary bit by XOR of the preceding binary bit with the current Gray bit:

Gray	1	1	0	0	1	1	0	1	0	1
Binary	1	0	0	0	1	0	0	1	1	0

Thus 1000100110_2 .

2. System and utility software

System software manages hardware and supplies services on which applications run: an operating system, device driver or language translator is an example. Utility software maintains, protects or optimizes a computer: backup tools, disk cleanup and antivirus tools are examples. Utilities are commonly treated as a subclass of system software, so the categories need not be mutually exclusive.

3. Microcomputers and minicomputers

Traditionally a microcomputer uses a microprocessor CPU and is designed primarily for one user's interactive work (a desktop PC, for example). A minicomputer is a historically intermediate multiuser system, with greater shared I/O and memory resources and support for multiple terminals under a time-sharing operating system (for example, a DEC PDP-11 system). Modern hardware blurs these historical boundaries; a microprocessor-based server can also serve many users.

4. Truth table

+ means OR and multiplication means AND.

A	B	C	$B + C$	$Y = A(B + C)$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

5. Translators

An assembler translates assembly-language mnemonics into machine/object code. A compiler translates a source program (or compilation unit) into target code before that target code runs; C compilation is an example. An interpreter executes a program through an interpreter runtime rather than requiring a separate native executable first. Compilation can detect syntax and type errors before target-code execution, but does not guarantee freedom from runtime errors. An interpreter may also parse or check code before execution, so saying that it always translates exactly one line at a time is an oversimplification. Compiled object code may require a linker and libraries; interpreted execution requires its runtime.

Real implementations can combine interpretation, bytecode compilation and just-in-time compilation; the distinction describes execution strategies rather than an absolute property of a language.