DAV PUBLIC SCHOOLS, ODISHA ZONE
HALF YEARLY EXAMINATION: 2023-24
CLASS :XI , SUBJECT :COMPUTER SCIENCE
BLUE PRINT OF QUESTION PAPER

Sl No.	Chapters / units	Marks Allotted in Syllabus	LA -II(5 marks) (3 nos)	LA -I(4 marks) (2 nos)	SA-II (3 marks) (5 nos)	SA-I (2 marks) (7 nos)	VSA (1 mark) (18 nos)	TOTAL (70 marks) (35 nos)
1	Computer System and Organisation	15				4	7	15
2	Computational Thinking and Programming -1	15		1	1	1	6	15
3	Flow of Control	25	1	1	3	1	5	25
4	String Manipulation	15	2		1	1		15
	Total		**3x5=15**	**2x4=08**	**5x3=15**	**7x2=14**	**18x1=18**	**70**

Subject: Computer Science   **Class : XI**   **Full Mark : 70**   **Nos. of Questions : 35**
As per the syllabus the typology of question as follows:

R →Remembering 10 % of 70 marks: (7 Marks)
U →Understanding 25 % of 70 marks: (18 marks)
A→Application 35% of 70 marks: (25 marks)
H →Higher Order Thinking 15% of 70 marks: (10 marks)
E →Evaluation 15% of 70 marks: (10 marks)

DAV PUBLIC SCHOOLS, ODISHA, ZONE

HALF YEARLY EXAMINATION: 2023-24
CLASS :XI , SUBJECT :COMPUTER SCIENCE

QUESTIONWISE ANALYSIS

Sl No.	Chapters / units	Forms of Question - (LA-I , LA-II,SA-II, SA-I, VSA)	Marks Allotted	(R), (U), (A), (H), (E)
1	Computer System and Organisation	SA-I-(4) VSA -(7)	15	(R)- 4 (U)- 5 (A) -3 (H)- 1 (E) -2
2	Computational Thinking and Programming-1	SA-I-(1) SA-II-(1) LA-I-(1) VSA -(6)	15	(R) -1 (U) -5 (A) -3 (H)- 3 (E) - 3
3	Flow of Control	LA-I-(1) LA-II (1) SA-I (1) SA-II-(3) VSA-(5)	25	(R)- 1 (U)- 5 (A) -13 (H)- 3 (E) - 3
4	String Manipulation	LA-II(2) SA-I-(1) SA-II-(1)	15	(R) -1 (U) -5 (A) -5 (H)- 1 (E) -3

DAV PUBLIC SCHOOLS, ODISHA, ZONE
HALF YEARLY EXAMINATION: 2023-24
CLASS : XI, SUBJECT :COMPUTER SCIENCE

MARKING SCHEME

QSTN NO	Value Points SECTION-A	Marks Allotted	TOT	PAGE NO. OF NCERT /TEXT BOOK
Q1.	a.ALU	1X18	18	P15-50
Q2.	b. 3000			
Q3.	d. NOR			
Q4.	d. Compiler			
Q5.	b. 10001100			
Q6.	b. -244.0			P160-200
Q7.	d. complex			
Q8.	c.Keywords			
Q9.	b. 10			
Q10	c.'computer'			275-290
Q11.	b. Hello			
Q12.	d.Or here?Or over here?			
Q13.	a. 5 times **or** d. (error)			P-373-383
Q14.	a. (X+Y'Z)			
Q15.	d.20 60			
Q16.	$(1222)_{10}$, $(2306)_8$ $(4C6)_{16}$			
Q17.	d. A is False but R is true			
Q18.	a. Both A and R are true and R is the correct explanation for A			
	SECTION B			
Q19.	UTF-8 is a variable-length character encoding standard used for electronic communication. Defined by the Unicode Standard, the name is derived from Unicode UTF-32 (32-bit Unicode Transformation Format) is a fixed-length encoding used to encode Unicode code points that uses exactly 32 bits (four bytes) per code	½ +½ ½ +½	2	P-15-20
Q20.	i) $(37365)_8$ ii) 11CC9	1 1	2	P-24-25
Q21	For Correct logic circuit diagram with expression **(OR)** A'B+AB+AB' (No partial Marking)	1+1 1	2	P-51 62

Q22.	Utility software helps users configure, analyse, optimise and maintain a computer. This software normally consists of small programs which are thought of as part of the operating system (OS) as they often come built in with the OS. Ex . (Text Editor , Antivirus)	2	2	P-8 11
Q23.	i) A logical error in a program is an error were the instructions given in the program do not accomplish the intended goal. "Get me a cup of coffee." is a logical error when the person intended to ask for a cup of tea. Ex. ctr=11 while ctr<=10: ii) The Python Value Error is raised when the wrong value is assigned to an object. This can happen if the value is invalid for a given operation, or if the value does not exist. For example, if a negative integer is passed to a square root operation, a ValueError is raised.	1 1	2	P-330-335

Q24.	Val = 25
	 for I in the range(0,Val) :
	    if I%2 == 0:
	      print(I+1)
	 else:
	      print (I-1)
	
	 (½ mark for each correct correction made and underlined.)
		½x4	2	

Q25.	It t	2	2	
	SECTION-C			

Q26.	hALF#yEARLY# 202
	                 1 mark for hALF#
	                 1 mark for yEARL
	                 1 mark for Y# 202
		1x3	3	P60-70

Q27.	name=input("Enter name")
	 amount=float(input("Enter amount of purchase"))
	 Type=input("Enter type of purchase(L/D)")
	 if Type=='L':
	    if amount<=25000:
	      D=0.0
	    elif amount<=50000:
	      D=5.0
	    else:
	      D=7.5
	 elif Type=='D':
	    if amount<=25000:
	      D=5.0
	    elif amount<=50000:
	      D=7.5
	    else:
	       D=10.0
	 else:
	    print("Enter the type as L or D only")
	 print("Net amount=",amount-(amount*D)/100)
		½
		
		 1
		
		 1
		
		 ½
			3	P-211 235

Q28.	Hi Everyone –Infinite times (with reason)	3	3	

	(OR)			
	1 3 5 7 9 11 13 15 17 19 (First 5 numbers carries 1½ ,next 6 numbers along with reason 1½)	3		
Q29.	y = float(input("How many years? ")) d = y * 365 h = d * 24 m = h * 60 s = m * 60 print(y, "years is:") print(d, "days") print(h, "hours") print(m, "minutes") print(s, "seconds")	½ ½ ½ ½ 1	3	

Q30.	import math
	 num=int(input("Enter a number = "))
	 a=int(math.sqrt(num))
	 b = 0
	 for i in range(2,a):
	   if a % i == 0 :
	      b = b + 1
	 if b == 0:
	      print(a,"Is a prime number ")
	 else:
	      print(a,"Is not prime number ")
		
		½
		½
		
		½
		½
		
		½
		½
			3	P-275
				280

SECTION-D				

Q31.	a) i=2
	    while i<20:
	        if i%3==0:
	            print(i)
	        i=i+1
		½
		½
		½
		½
			5	P-278

	b) x = int(input("Enter the base = "))
	   fact = 1
	   sum = 0
	   for i in range(1,7):
	     fact = fact * i
	     if i % 2 == 0 :
	       sum = sum - (x**i)/fact
	     else :
	       sum = sum + (x**i)/fact
	   print("Sum of series = ",sum)
	
	(Any other alternative method may be considered)
	
	(OR)
		
		½
		
		½
		½
		½
		
		½
		½
				P-272
				289

	a) 14	2		

	b)
	even_total = 0
	odd_total = 0
	negtot=0
	while (number != 0):
	  number=int(input(" Please Enter the Number : "))
	  if(number<0):
	    negtot = negtot + number
	  elif (number %2==0):
	    even_total = even_total + number
	  else:
	    odd_total = odd_total + number
	print("The Sum of Even Numbers ", even_total)
	print("The Sum of Odd Numbers ", odd_total)
	print("The Sum of -Ve Numbers ", negtot)
		½
		
		½
		
		½
		½
		
		1		

Q32	a) 1. string.partition(sep/string)
	
	   2. find() returns the lowest index in the string where the substring sub is found within the slice s[start:end] . It returns -1 if the sub is not found.
	
	b)s=input("Enter a string:")
	 t=""
	 for i in s:
	  if i.lower() in ['a','e','i','o','u']:
	   t=t+chr(ord(i)+1)
	  else:
	   t=t+i
	 print("Original string=",s)
	 print("New string=",t) (Any Other logic may be considered)
		1
		
		1
		
		½
		½
		½
		½
		
		½
		½
			5	P370
				376

Q33	S = input('Enter a sentence : ')
	number_of_words = 1
	number_of_characters = len(S)
	al_num = 0
	for i in S:
	 if i.isalnum():
	  al_num += 1
	 if i == ' ' : # there is a space means there is another word
	  number_of_words += 1
	
	a) print('Number of words are', number_of_words)
	b) print('Number_of_characters are', number_of_characters)
	
	          (OR)
	b) print('Percentage of characters that are alphanumeric is', al_num*100/len(s), ' %')
		½
		½
		½
		½
		½
		½
		½
		½
		½
		½
		
		½
		½
			5	P-380
				390

	SECTION-E			
Q34.	(i) float( ) (ii) input("Enter an operator [+ - * / %] :") (iii) Invalid operator!! (iv) His code becomes less efficient from execution point of view.	1 1 1 1	4	P-276-280
Q35.		½ ½ ½ ½ ½ ½ ½ ½	4	P-335-345