KENDRIYA VIDYALAYA SANGATHAN, MUMBAI REGION

2ND PRE- BOARD EXAMINATION 2018-19

Subject: Computer Science (083)

CLASS:12th (Twelfth)

Duration: 3Hrs **Max Marks: 70**

MARKING SCHEME

Q.No		Answer	Marks Distribution
1	a)	• Correct role • Yes with justification	1 ½+½
	b)	iostream.h math.h	½ + ½
	c)	• Header file stdio.h to be included • Colon (:) after public is missing • Semicolon (;) at the end of class missing • E1.getexamdetails() • E1.showexamdetails	½ Mark for each identified error. 1/2X4
	d)	o9 D8 y7 n6	½ Mark for each correct line of output
	e)	20, 23 10 ,23 11 ,11	1 Mark for each correct line of output
	f)	Option (iv)	2 Marks
2	a)	If the visibility mode of the base class is private i.e. if the base class has been privately derived then the public and protected members of the base class become private members of the derived class. if the visibility mode of the base class is protected i.e. if the base class has been protectedly derived then the public and protected members of the base class become protected members of the derived class.	2 Mark for correct difference with example
	b)	(i) constructor overloading/polymorphism/function over loading. (ii) destructor, invoked when object goes out of scope **OR** 1 mark for definition of copy constructor 1 mark for correct Example of copy constructor	1 Mark each
	c)	class show { char name_of_show[20]; char date_of_release[20], name_of_director[20]; int star;	(½ Mark for correct syntax for class header) (½ Mark for correct declaration

		inttotal_print_release;
		public:
		show()
		{
		strcpy(Nameofshow," ");
		strcpy(dateof release,"1_1_2007");
		strcpy(nameof director," ");
		star=2;
		totalprintrelease=100;
		}
		calculate_star()
		{
		if(totalprintrelease>=1000)
		star =5;
		else if(totalprintrelease>=500)
		star=4;
		else if(totalprintrelease>=300)
		star=3;
		else if(totalprintrelease>=100)
		star =2;
		else
		star =1;
		}
		void EnterShow()
		{
		gets(name_of_Show,);
		gets(date_of_release);
		gets(name_of_director);
		cin>>total_print_release;
		calculate_star();
		}
		
		void DisplayShow()
		{
		cout<<name_of_Show<<date_of_release;
		cout<<name_of_director<<total_print_release<<Star;
		}
			of data members)
			(½ Mark for constructor)
			(1 Mark for calculation of correct star for each condition)
			(1 Mark for correct definition of EnterShow() with proper invocation of calculate_star())
			(½ Mark for correct definition of function DisplayShow())

	d)	(i) Hierarchical inheritance (ii) Location, Area, Sale (iii) Enter(), Show(), Register(), Show() (iv) Input(),Output() **OR** class MEDIA : public PRINT, private DIGITAL { intMID; char Mtype[20]; char Mname[20]; public: MEDIA() { MID=1; strcpy(Mtype,"NULL");	1 1 1 1 **OR** (1 Mark for correct syntax for derived class header) (½ Mark for correct declaration of data members) (½ Mark for defining

		strcpy(Mname,"NULL");
		}
		void Enter( )
		{
		cin>>MID;
		gets(Mtype);
		gets(Mname);
		}
		void Display( )
		{
		cout<<MID;
		cout<<Mtype;
		cout<<Mname;
		}
		};
			constructor MEDIA( ))
			
			(1 Mark for defining the function Enter( ))
			(1 Mark for defining the function Display( ))

3	a)	void SumEO(int VALUES[], int N)
		{int SE = 0, SO = 0;
		for (int I=0;I<N;I++)
		{
		if(VALUES[I] %2 == 0) SE += VALUES[I];
		else
		SO += VALUES[I];
		}
		cout<< "Sum of even values = " << SE<<endl; cout<< "Sum of odd values = " << SO<<endl;
		}
		OR
		void MIXER (int A[], int N)
		{
		for (int I=0;I<N;I++)
		{
		if(A[I] %2 != 0)
		A[i] *=2;
		else
		A[i] *=3;
		}
		}
		
		Any other correct alternative code in C++
			(½ Mark for correctly writing the loop)
			(½ Mark for adding even elements)
			(½ Mark for adding odd elements)
			(½ Mark for displaying the sum of even and odd elements)

	b)	Given, W=2 N=40 M=30 Base(S)=5000 Row Major Formula: Loc(S[I][J]) =Base(S)+W*(M*I+J) Loc(S[20][10]) =5000+2*(30*20+10) =5000+2*(600+10) =5000+1220 =6220	*(1 Mark for writing correct formula (for column major) OR substituting formula with correct values)* *(1 Mark for writing calculation step - at least one step)* *(1 Mark for correct address)*

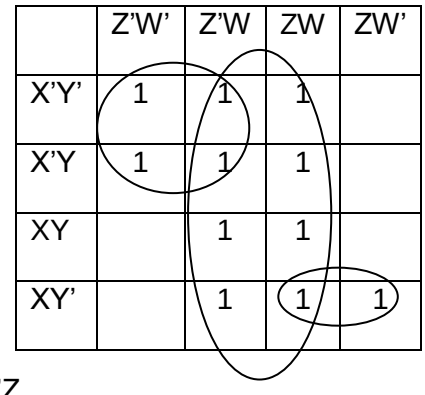
		OR Base Address=50792 Address of G[5] [10]=51412	

	c)	void UpperHalf(int Mat[4][4])
		{ for (int I=0;I<4;I++)
		{ for (int J=0;J<4-I;J++)
		cout<<MAT[I][J]<< " " ;
		cout<<endl;
		}}
		OR
		void TOPBOTTOM(int M[][5],intR,intC)
		{intsumtop=0,sumbottom=0;
		for (int I=0;I<R;I++)
		{ for (int J=0;J<C;J++)
		{
		If(i==0)
		Sumtop=sumtop+M[i][j];
		If(i==R-1)
		Sumbottom=sumbottom+M[i][j];
		}
		}
		cout<<"sumtop="<<sumtop<<",sumbottom="<<sumbottom;
		}
			(½ Mark for correctly writing loop for traversing rows)
			(½ Mark for correctly writing loop for traversing columns in each row)
			(1 Mark for correctly checking elements for display)
			(½ Mark for correctly displaying the selected elements)
			(½ Mark for correctly displaying line break after each row)

	d)	void insert()
		{
		If(rear= = size-1 && front ==0|| front=rear+1)
		{
		cout<<"queue full….";
		return;
		}
		else if(rear= = -1)
		{
		rear++;
		front++;
		} else if(rear==size-1)
		rear=0;
		else
		{
		rear++;
		//add element
		}
		OR
		(1 Mark for creating new node Book)
		(1 Mark for assigning top to temp)
		(1 Mark for top=top->next) (1 Mark for insert node)
			4 marks for any correct logic

	(e)	1 mark for showing step by step stack contents 1 mark for final answer **OR**	1+1

		1 mark for showing step by step stack contents 1 mark for final answer	
4	a)	File.seekg(0,ios::end) ; File.tellg() ; **OR** 1 mark for correct difference	½ + ½
	b)	• 1 mark for correct opening of file • 1 mark for correct logic for counting lowercase alphabets **OR** • 1 mark for correct opening of file • 1 mark for correct logic for counting number of lines	1+1
	c)	• For correct opening of file 1mark • For selecting correct record 1 mark • For writing to backup file 1mark **OR** • (½ Mark for opening file correctly) • (1 Mark for reading each record from file) • (1 Mark for comparing value) • (½ Mark for displaying thematching record)	1+1+1
5	a)	Degree – No.of columns present in a table Cardinality – No.of rows present in a table	1+1
	b)	(1) (i) select * from WORKERS ORDER BY DOJ DESC; (ii) select NAME,DESIGNATION FROM WORKERS where PLEVEL IN('P001', 'P002'); (iii) select * from WORKERS where DOB between '10-Mar-1984' and '31-Dec-1986'; (iv) Select ECODE, NAME , PAY from WORKER, PAYLEVEL where WORKER.PLEVEL=PAYLEVEL.PLEVEL and DESIGNATION=**'** Operator'; (2) (i) PLEVEL count(*) P001 1 P002 2 P003 2 (ii) Max(DOB) Min(DOB) 12-July-1987 23-Aug-1981 (iii) NAME PAY Deepak24000 Dharmend 24000 (iv) PLEVEL PAY+ALLOWANCE P002 34000	Q1. 1+1+1+1 Q2 ½ + ½ + ½ + ½
6	a)	Absorption law i: X+XY=X ii: X(X+Y) = X i. X Y XY X+XY 0 0 0 0 0 1 0 0 1 0 0 1 1 1 1 1	(1 mark for correct statement) (1 mark for proof of any one Absorption law)

<table>
<tr><td></td><td></td><td>ii.
X Y X+Y X(X+Y)
0 0 0 0
0 1 1 0
1 0 1 1
1 1 1 1
Hence proved.</td><td></td></tr>
<tr><td></td><td>b)</td><td>F= (A'.B)+(A.B)+(B'.C)</td><td>2</td></tr>
<tr><td></td><td>c)</td><td>G= P'QR'+PQ'R'+PQR'</td><td>1</td></tr>
<tr><td></td><td>d)</td><td>F (X, Y, Z, W) = Σ (0,1,3,4,5,7,9,10,11,13,15)
 
F= W+X'Z'+XY'Z</td><td>(1 mark for correct numbering)
(1 mark for correct grouping)
(1 mark for writing correct expression)</td></tr>
<tr><td>7</td><td>a)</td><td>Correct difference</td><td>1+1</td></tr>
<tr><td></td><td>b)</td><td>GSM- Global System for Mobile Communication
CDMA- Code Division Multiple Access
WLL- Wireless in Local Loop
SMTP –Simple Mail Transfer Protocol</td><td>½ each</td></tr>
<tr><td></td><td>c)</td><td>(i) Any possible cable layout
(ii) Gamma, More number of computers
(iii) Repeater- Justification with cable type used
Hub/switch – in all buildings
(iv) Any wireless technology can be given 1 Mark</td><td>1+1+1+1</td></tr>
<tr><td></td><td>d)</td><td>Any two example</td><td>½ + ½</td></tr>
<tr><td></td><td>e)</td><td>URL= "http://www.XtSchool.com/default.aspx"
Domain name= XtSchool.com</td><td>½ + ½</td></tr>
<tr><td></td><td>f)</td><td>Cookies: are the messages that a web server transmits to a web browser so that the web server can keep track of the user's activityon a specific website.
Firewall: is a system designed to prevent unauthorized access to or from a private network.</td><td>½ + ½</td></tr>
</table>