SECTION A

[Only for candidates, who opted for C++]

1. (a) Write the names of any four fundamental data types of C++. *2*

(b) Write the names of the correct header files, which must be included in the following C++ code to compile the code successfully : *1*

```
void main()
{
   char L[]="CS 2018";
   int N=strlen(L);
   cout<<L[N-1];
}
```

(c) Rewrite the following **C++ program** after removing any/all syntactical error(s). Underline each correction done in the code : *2*

Note : Assume all required header files are already included in the program.

```
#define Area(L,B) = L*B
structure Recta
{
   int Length,Breadth;
};
void main()
{
   Recta R = [10,15];
   cout<<Area(Length.R,Breadth.R);
}
```

(d) Find and write the output of the following C++ program code : *2*

Note : Assume all required header files are already included in the program.

```
void Alter(char *S1, char *S2)
{
  char *T;
  T=S1;
  S1=S2;
  S2=T;
  cout<<S1<<"&"<<S2<<endl;
}
void main()
{
  char X[]="First", Y[]="Second";
  Alter(X,Y);
  cout<<X<<"*"<<Y<<endl;
}
```

(e) Find and write the output of the following C++ program code : 3

Note : Assume all required header files are already included in the program.

```
void Convert(float &X, int Y=2)
{
   X=X/Y;
   Y=X+Y;
   cout<<X<<"*"<<Y<<endl;
}
void main()
{
   float M=15, N=5;
   Convert(M,N);
   Convert(N);
   Convert(M);
}
```

(f) Observe the following C++ code and find the possible output(s) from the options (i) to (iv) following it. Also, write the minimum and maximum values that can possibly be assigned to the variable **End**. 2

Note :

- Assume all the required header files are already being included in the code.
- The function random(N) generates any possible integer between 0 and N-1 (both values included).

```
void main()
{
  randomize();
  int A[]={10,20,30,40,50,60,70,80};
  int Start = random(2) + 1;
  int End = Start + random(4);
  for(int I=Start; I<=End, I++)
     cout<<A[I]<<"$";
}
```

(i) **10$20$30$**	(ii) **20$30$40$50$60$**
(iii) **30$40$50$60$**	(iv) **40$50$60$70$**

2. (a) Given the following class Test and assuming all necessary header file(s) included, answer the questions that follow the code :

```
class Test
{
   int Marks; char TName[20];
public:
   Test(int M)                  //Function 1
   {
      Marks = M;
   }
   Test(char S[])               //Function 2
   {
      strcpy(TName,S);
   }
   Test(char S[], int M)        //Function 3
   {
      Marks = M;
      strcpy(TName,S);
   }
   Test(Test &T)                //Function 4
   {
      Marks = T.Marks + 10;
      strcpy(TName,T.TName);
   }
};

void main()
{
   Test T1(10);                   //Statement I
   Test T2(70);                   //Statement II
   Test T3(30, "PRACTICAL");      //Statement III
   _______________;               //Statement IV
}
```

(i) Which of the statement(s) out of (I), (II), (III), (IV) is/are incorrect for object(s) of the class Test ? *1*

(ii) What is Function 4 known as ? Write the **Statement IV**, that would execute **Function 4**. *1*

(b) Observe the following C++ code and answer the questions (i) and (ii).

Note : Assume all necessary files are included.

```
class Point
{
   int X,Y;
public:
   Point(int I=10, int J=20)   //Function 1
   {
       X = J;
       Y = I;
   }
   void Show()                 //Function 2
   {
       cout<< " Points are " << X << " & " << Y <<endl;
   }
   ~Point()                    //Function 3
   {
       cout<<"Points Erased"<<endl;
   }
};

void main()
{
   Point P(5);
   P.Show();
}
```

(i) For the class Point, what is **Function 3** known as ? When is it executed ? *1*

(ii) What is the output of the above code, on execution ? *1*

OR

(b) Explain Polymorphism in context of Object Oriented Programming. Also give a supporting example in C++. *2*

(c) Write the definition of a class GRAPH in C++ with following description : *4*

Private Members

- **XUnit** // integer
- **YUnit** // integer
- **Type** // char array of size 20
- **AssignType()** /* Member function to assign value of Type based upon XUnit and YUnit as follows : */

Condition	**Type**
XUnit = 0 Or YUnit = 0	None
XUnit is more than YUnit	Bar
XUnit is less than or equal to YUnit	Line

Public Members

- **InXY()** /* Function to allow user to enter values of XUnit and YUnit and then invoke AssignType() to assign value of Type */
- **OutXY()** // Function to display XUnit, Yunit and Type

(d) Answer the questions (i) to (iv) based on the following : *4*

```
class Ground
{
   int Rooms;
protected:
   void Put();
public:
   void Get();
};

class Middle : private Ground
{
   int Labs;
public:
   void Take();
   void Give();
};

class Top : public Middle
{
   int Roof;
public:
   void In();
   void Out();
};

void main()
{
   Top T;
}
```

(i) Which type of Inheritance out of the following is illustrated in the above example ?

– Single Level Inheritance, Multilevel Inheritance, Multiple Inheritance

(ii) Write the names of **all the members**, which are directly accessible by the member function **Give()** of class **Middle**.

(iii) Write the names of **all the members**, which are directly accessible by the member function **Out()** of class **Top**.

(iv) Write the names of **all the members**, which are directly accessible by the object **T** of class **Top** declared in the **main()** function.

OR

(d) Consider the following class HeadQuarter *4*

```
class HeadQuarter
{
   int Code;
   char Des[20];
protected :
   char Address[40];
public:
   void Get(){cin>>Code;gets(Des);gets(Address);}
   void Put(){cout<<Code<<Des<<Address<<endl;}
};
```

Write a code in C++ to protectedly derive another class FrontOffice from the base class HeadQuarter with following members.

Data Members

Location of type character of size 10

Budget of type double

Member Functions

A constructor function to assign Budget as 100000

Assign() to allow user to enter Location and Budget

Display() to display Location and Budget

3. (a) Write a user-defined function **NoTwoThree(int Arr[], int N)** in C++, which should display the value of all such elements and their corresponding locations in the array **Arr** (i.e. the array index), which are **not multiples of 2 or 3. N** represents the total number of elements in the array **Arr**, to be checked. *3*

Example : If the array Arr contains

0	1	2	3	4
25	8	12	49	9

Then the function should display the output as :

25 at location 0

49 at location 3

OR