# First Term Class XII CS Cycle #2 Weekly Test held on 14-05-2017 Solution

1. a) Explain the following concepts with example: i) Data Hiding ii) Polymorphism [3]

**Data Hiding**: Keeping the data in private or protected area of a class to prevent it from accidental modification (change) from outside the class is known as data hiding. Private or protected data members can only be accessed by using public member functions of the same class.

```
class student
{
  char na[20]; double marks;
  public:
    void input() { gets(na); cin>>marks; }
    void display() { cout<<na<<","<<marks<<endl; }
    double retmarks() { return marks; }
    void update(double ma) { marks=ma; }
};
```

**Polymorphism:** The process of using an operator or a function in different ways for different set of inputs given is known as polymorphism, that is, functions / operators have same name but perform different actions depending on parameters.

```
void print(int i) { cout<<"Integer = "<<i<<endl; }
void print(double d) { cout<<"Double = "<<d<<endl; }
void print(char s[]){ cout<<"String = "<<s<<endl; }
```

b) Differentiate between access specifier protected and private. [2]

| Access Specifier Protected | Access Specifier Private |
|---|---|
| Member inherited from the base class become protected member in the derived class<br>Inherited member can be inherited | Member inherited from the base class become private member in the derived class<br>Inherited member cannot be inherited |

c) Mention the role of constructor and destructor of a class. Write any two similarities between constructor and destructor of a class. [2]

Constructor is a special public member of a class that creates an object. Destructor is special public member function of a class that destroys an object, when the object goes out scope.

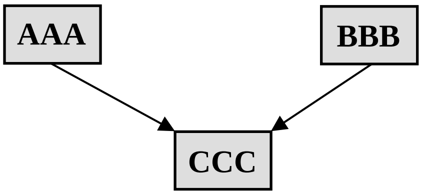
Two similarities between constructor and destructor of a class are (**select any two**):

1. Visibility label public 2. No return type, not even void 3. Cannot be inherited

d) With examples differentiate between multiple inheritance and multi-level inheritance. [2]

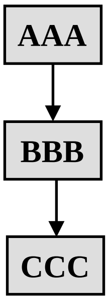
**Multiple Inheritance**: A new class is derived from more than two or more base classes.

```
class AAA { //Members of AAA };
class BBB { //Members of BBB };
class CCC: public AAA, public BBB
{
  //Members of CCC
};
```



**Multi-level Inheritance**: A new class is created from a derived class, that is, between top most base class and bottom most derived there is at least one intermediate class.

```
class AAA               //top most base class
{
  //Members of AAA
};
class BBB: public AAA  //intermediate class
{
  //Members of BBB
};
class CCC: public BBB  //bottom most derived class
{
  //Members of CCC
};
```



e) i) What is the difference between output mode and append mode when writing into a file? [1]

**Output mode**: if a file already exists, new set of data will overwrite existing data in the file, that is, there will be data loss

**Append mode**: if a file already exists, new set of data will be added at the end of the file, that is, there will be no data loss

ii) What is the difference between >> and get() when reading from a text file? [1]

>> (**extraction operator**): will read a text file character by character but **ignores** white space characters like space, tab and new line characters

**get()**:will read a text file character by character **including** white space characters like space, tab and new line characters

2. Find output of the following program segment: [2]

```
#include <iostream.h>
class place
{
  int temp, humi, polu;
  public:
    place(int t=2) { temp=t; humi=60; polu=50; }
    void edit(int x) { humi-=x; polu+=x; }
    void modify(int y) { humi+=y; polu-=y; }
    void show() { cout<<temp<<":"<<humi<<"&"<<polu<<endl;}
};
void main()
{
  place a, b(5);
  a.edit(8);
  b.modify(7);
  a.show();
  b.show();
  b.edit(10);
  a.modify(-5);
  b.show();
  a.show();
}
```

```
2:52&58
5:67&43
5:57&53
2:47&63
```

3. Assume all the header files are included in the source code. Rewrite the complete program after removing all the syntax errors (**Underline the corrections**): [2]

```
//Writes into a text file
void main()
{
  ofstream textfile("SPORTSDAY.TXT");
  //Or, fstream textfile("SPORTSDAY.TXT", ios::out);
  char line[80];
  for (int x=0; x<10; x++)
  {
    cout<<"Input a line of text? "; gets(line);
    textfile<<line<<endl;
  }
  textfile.close();
}
```

4. Declare a class `employee` with the following members (define all the member functions): [4]

**Private members**:

`code` integer (employee code)
`name` string of 20 characters (employee name)
`bsal` double (basic salary per month)
`hra` double (house rent allowance per month = 40% of `bsal`)
`gsal` double (gross salary per month = `bsal`+`hra`)
`itax` double (income tax)
`nsal` double (net salary per month = `gsal-itax`)
`salary`() function to calculate `hra`, `gsal`, `itax` and `nsal`. Income Tax (`itax`) is calculated as:

| Gross salary | Income Tax |
|---|---|
| <=200000 | 0 |
| >200000 and <=500000 | 5.0% of Gross Salary |
| >500000 and <=1000000 | 7.5% of Gross Salary |
| >1000000 | 10.0 % of Gross Salary |

**Public members**:

Default constructor to initialize `hra`, `gsal`, `itax` and `nsal` to `0` (zero).
Destructor to display "Employee is on long vacation".
`empinput`() function to accept values from keyboard for `code`, `name` and `bsal`.
`empshow`() function to call `salary`() and display all the data members on the screen.

```
class employee
{
  int code; char name[20];
  double bsal, hra, gsal, itax, nsal;
  void salary()
  {
    hra=0.4*bsal; gsal=bsal+hra;
    if (gsal>=200000 && gsal<500000)
      itax=0.05*gsal;
    else
    if (gsal<1000000)
      itax=0.075*gsal;
    else
    if (gsal>1000000)
      itax=0.10*gsal;
    nsal=gsal-itax;
  }
  public:
    employee() { hra=gsal=itax=nsal=0; }
    ~employee() { cout<<"Employee is on long vacation\n"; }
    void empinput()
    {
      cout<<"Code? "; cin>>code;
      cout<<"Name? "; gets(name);
      cout<<"BSal? "; cin>>bsal;
    }
    void empshow()
    {
      salary();
      cout<<"Code="<<code<<endl;
      cout<<"Code="<<code<<endl;
      cout<<"Name="<<name<<endl;
```

```
        cout<<"BSal="<<bsal<<endl;
        cout<<"HRA ="<<hra<<endl;
        cout<<"GSal="<<gsal<<endl;
        cout<<"ITax="<<itax<<endl;
        cout<<"NSal="<<nsal<<endl;
      }
  };
```

5. Consider the following class declarations and answer the questions given below: [4]

```
class AAAA
{
  int x1, x2; void a1();
  protected: char x3[20]; void a2();
  public: double x4; AAAA(); void inputa(); void showa();
};
class BBBB
{
  double y1, y2; void b1();
  protected: char y3[40]; void b2();
  public: int y4; BBBB(); void inputb(); void showb();
};
class CCCC: private BBBB, public AAAA
{
  char z1[20]; void c1();
  protected: float z2, z3; void c2();
  public: double z4; CCCC(); void inputc(); void showc();
};
```

a) Name the private member functions of the class CCCC.
void b2(), void c1(), void inputb(), void showb()

b) Name the inherited data members of the class CCCC.
char x3[20], double x4, char y3[40], int y4

c) Name the member functions that can be used with an instance of CCCC.
void inputb(), void showb(), void inputc(), void showc();

d) How many bytes of memory is allocated to an instance of BBBB and to an instance of CCCC.
Number of bytes allocated to BBBB is 60, number bytes allocated to CCCC is 132

6. a) Write a C++ function to read a text file "PLAY.TXT"; count and display number of vowels present in the file. [2]

```
void qno6a()
{
  ifstream fin("PLAY.TXT");
  char ch; int count=0;
  while (fin.get(ch))
    if (ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U' ||
      ch=='a' || ch=='e' || ch=='i'|| ch=='o' || ch=='u')
      count++;
  /*
    ch=toupper(ch);
    if (ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')
    Or,
    ch=tolower(ch);
    if (ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
  */
```

```
    fin.close();
    cout<<"Vowels="<<count<<endl;
}
```

b) Write a C++ function to read a text file "PLAY.TXT"; count and display number of times "THEN" appear in the file. [2]

```
void qno6b()
{
    ifstream fin("PLAY.TXT");
    char word[15]; int count=0;
    while (fin>>word)
        if (strcmpi(word, "THEN")==0)
            count++;
    fin.close();
    cout<<"Number of times 'THEN' appear in the file="<<count<<endl;
}
```

c) Write a C++ function to read a text file "PLAY.TXT"; count and display number lines starting with 'T' / 't' in the text file. [2]

```
void qno6c()
{
    ifstream fin("PLAY.TXT");
    char line[80]; int count=0;
    while (fin.getline(line, 80))
        if (line[0]=='T' || line[0]=='t')
            count++;
    fin.close();
    cout<<"Number of line starting wiht 'T'/'t'="<<count<<endl;
}
```

7.
```
class car
{
    char model[30]; double miles, price;
    public:
        void cinput()
        {
            cout<<"Model? "; gets(model);
            cout<<"Miles Price? "; cin>>miles>>price;
        }
        void cdisplay() { cout<<model<<","<<miles<<","<<price<<endl; }
        double retmiles() { return miles; }
        double retprice() { return price; }
};
```

a) Write a function in C++ to append n records (n is a parameter) in the binary file "CARS.DAT" containing objects of the type `car`. [3]

```
void qno7a(int n)
{
    ofstream fout("CARS.DAT", ios::binary|ios::app);
    car c;
    for (int k=0; k<n; k++)
    {
        c.cinput(); fout.write((char*)&c, sizeof(c));
    }
    fout.close();
}
```

b) Write a function in C++ to read objects from a binary data file "CARS.DAT" containing objects of the type `car`. Display only those records where miles<50000 and price<2 lakhs. If no such record is found then display an error message. [3]

```
void qno7b()
{
  ifstream fin("CARS.DAT", ios::binary);
  car c; int count=0;
  while (fin.read((char*)&c, sizeof(c)))
    if (c.retmiles()<50000 && c.retprice()<200000)
    {
      c.cdisplay();
      count++;
    }
  fin.close();
  if (count==0)
    cout<<"No objects/records found in the file\n";
}
```

8. a) MAT[30][25] is a 2D array stored in the computer's main storage. Base address of MAT is 2000 and every element of MAT is allocated 8 bytes. Find the address of MAT[20][15] when MAT is stored as i) Row Major ii) Column Major [3]

**Row Major** : b+w*(r*NOC+c) = 2000+8*(20*25+15) = 6120

**Column Major** : b+w*(r+c*NOR) = 2000+8*(20+15*30) = 5760

b) Give the output of the program given below: [2]

```
#include<iostream.h>
void main()
{
  int arr[]={39, 57, 86, 48, 67, 78, 48, 97};
  int *ptr=arr, s1=0, s2=0;
  for (int k=0; k<8; k++, ptr++)
  {
    s1+=*ptr/10;
    s2+=*ptr%10;
    if (k%2==0)
      cout<<s1<<","<<s2<<endl;
  }
}
```

```
3,9
16,22
26,37
37,53
```