(c) Write a function UpperTriangle() that accepts a square matrix A and its order n as input arguments. The function should convert matrix A to an upper triangular matrix by assigning 0 to all elements below the diagonal (diagonal left to right from top). (5)

[This question paper contains 16 printed pages.]

Your Roll No.....

Sr. No. of Question Paper : 1197 **F**

Unique Paper Code : 2342011201

Name of the Paper : Object-Oriented Programming with C++ (DSC04)

Name of the Course : **B.Sc. (H) Computer Science**

Semester : II

Duration : 3 Hours Maximum Marks : 90

Instructions for Candidates

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. **Section A** is compulsory (Question **1**).
3. Attempt **any 4** questions from **Section B** (Questions **2** to **6**).

Section A

(Compulsory Question)

1. (a) What are inline functions? Rewrite the following code using the inline function. (3)

```
#include<iostream>
using namespace std;
float mul (int x, int y)
{
    return (x*y);
}
int main()
{
    int a = 2, b = 5;
    cout << mul(a, b) << "\n";
    return 0;
}
```

(b) What will be the output of the following program :

(i) (3)

```
#include<iostream>
using namespace std;
class construct
{
int p, q;
```


5. (a) Write a program to define a class, Complex, with the following features : (10)

   (i) data members hidden from outside the class

   (ii) a default and parametrised constructor

   (iii) a member function to add another complex number to it main() function to show the implementation of the class

   (b) Write a function that compares the two given arrays arr1 and arr2 of the same size (passed as parameters) for equality, and returns true or false. (5)

6. (a) What is a pure virtual function? Define an abstract class Polygon, with a data member area that stores the area of the Polygon, and a pure virtual function that calculates the area of the Polygon. Inherit a Rectangle class from the Polygon. Complete the program to show the use of the abstract class and polymorphism. *(10)*

```
};
int main()
{
        basel *ptr;
        der x;
        ptr = &x;
        ptr->print();
        ptr->show();
}
```

4. (a) Write a program to print the following output : (6)

```
1
12
123
1234
12345
.....
```

(b) Write a program to print the area of a square and circle using function overloading. (9)


```
public:
  construct(int x, int y)
  {
     p = x;
     q = y;
  }
  void Display()
  {
     cout<<p<<"\n"<<q<<"\n";
  }
};
int main()
{
  construct item1(10, 20), item2 =
construct(30, 40);
  item1.Display();
  item2.Display();
  return 0;
}
```

(ii) (3)

```
#include<iostream>
using namespace std;
void square(int* snum)
{
   cout<<"Square of 10 is ";
   *snum *= *snum;
}
int main()
{
   int num = 10;
   square (&num);
   cout << num << end1;
}
```

(iii) (3)

```
#include<iostream>
using namespace std;
void Myclass()
{
   try
   {
      throw “y”;
```



```
      cout<<"print version of base class"<<end1;
   }
   void show()
   {
      cout<<"Show version of base class"<<end1;
   }
};
class der: public basel
{
   public:
      void print()
      {
         cout << "print version of derived class " <<
end1;
      }
      void show()
      {
         cout << "Show version of derived class" <<
end1;
      }
```

3. (a) Create a class ThreeDim which contains x, y and z coordinates as integers. Define the following for the class :

(i) default constructor to initialize data members to zero

(ii) parametrized constructor to initialize data members to values passed

(iii) function out() to display the coordinates of the class. (9)

(b) What will be the change in the output if a virtual keyword is removed from the print () function of the class basel? Write the output for the following code with the virtual keyword and without it. (6)

```
#include<iostream.h>
using namespace std;
class basel
{  public:
    virtual void print()
    {
```



```
    }
        catch (const char*)
    {
        cout<<"Exception inside Myclass\n";
        throw;
    }
}
int main()
{
    cout<<"Now main starts\n";
    try
    {
        Myclass();
    }
    catch (const char*)
    {
        cout<<"Exception inside main\n" ;
    }
    cout<<"Now main ends\n";
    return 0;
}
```

(c) Write a program that takes a character from the keyboard and displays its corresponding ASCII value on the screen. (3)

(d) How do the properties of the following two derived classes A and B differ?

(i) class A: private C{//...};

(ii) class B: public C{//...}; (3)

(e) Write a function to swap two numbers using pointer datatype parameters. (3)

(f) Identify the error(s) in the following program :

(i) (3)

```
#include<iostream>
using namespace std;
class four_seater
{
    public:
        void Property()
        {
```



```
#include<iostream>
using namespace std;
int func(int a, int* b, int& c)
{
    int temp = a + *b + c;
    a += 10;
    *b += 20;
    c += 30;
    return temp;
}
int main()
{
    int x = 1, y = 2, z = 3;
    cout << x << ", " << y << ", " << z << "\n";
    cout << func(x, &y, z);
    cout << "\n" << x << ", " << y << ", " << z;
    return 0;
}
```

SECTION B

2. (a) Write a program that reads a text file and creates an output file, named "out. dat". The output file is identical to the text file except that every sequence of consecutive blank spaces is replaced by a single space. (5)

(b) What is the sequence of constructors and destructors being called in the following multilevel inheritance : (5)

```
class X
{...};
class Y: public X;
{...};
class Z: public Y;
{...};
```

(c) Write the output of the following code. Also, mention the call by value and call by reference parameters in the following code. (5)


```
        cout<<"It has space for four
        persons"<<end1;
    }
};
class four_wheeler
{
    public:
        void Property()
        {
            cout<<"It runs on four tyres"<<end1;
        }
};
class Car: public four_seater, public four_wheeler
{ };
int main ()
{
    Car C1;
    C1.four_seater;
    C2.four_wheeler;
    return 0;
}
```

(ii) #include<iostream> (3)

```
using namespace std;
Template<class T1, class T2>
class Person
{
    T1 m_t1;
    T2 m_t2;
public:
    Person (T1 t1, T2 t2)
    {
        m_t1=t1;
        m_t2=t2;
        cout<<m_t1<<" "<<m_t2<<end1;
    }
    Person (T3 t2, T4 t1)
    {
        m_t2=t2;
        m_t1=t1;
        cout<<m_t1<<" "<<m_t2<<end1;
    }
};
```



```
void main()
{
    Person <int, float> obj1(1, 2.34);
    Person <float, char> obj2(2.13, 'r');
}
```

(iii) # include <iostream> (3)

```
#include <fstream>
using namespace std;
int main()
{
    const int size = 100;
    char buffer[size];
    ifstream in ("p1.cpp");
    ofstream out("p2.cpp");
    while(in.get(buffer))
    {
        in.get();
        cout<<buffer<<end1;
        cout<<buffer<<end1;
    }
    in.close();
    out.close();
}
```