KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY

B.Sc. Engineering 2ndYear 1st Term Examination, 2016
Department of Computer Science and Engineering
CSE 2101
Object Oriented Programming

TIME: 3 hours FULL MARKS: 210

N.B. i) Answer **ANY THREE** questions from each section in separate scripts.
ii) Figures in the right margin indicate full marks.

SECTION A

(Answer **ANY THREE** questions from this section in Script A)

1. a) What is Object Oriented Programming (OOP)? Encapsulated classes are easier to change. Give an example with necessary code. (09)
   b) How can you access static variable of a class without creating any object? (05)
   c) Write a template function which returns the average of all the elements of an array of any built in data type. Extend your program for a user defined data type. (10)
   d) How does C++ handle different signals? Write the general syntax of signal() function. (06)
   e) How does a function return another function with the help of a function pointer? (05)

2. a) How does internal shared ownership counter work while working with shared pointer? Explain with necessary code. (06)
   b) Suppose you enter 5 elements in a vector. Use Lambda expression and find_if algorithm to search a particular element in that vector. (08)
   c) How can you make the member function of a class friend of another class? Explain with necessary code. (08)
   d) Write a class to represent a vector (a series of float values). Include member functions to perform the following tasks: (13)
      i) To create the vector
      ii) To modify the value of a given element
      iii) To multiply by a scalar value
      iv) To display the vector in the form (10, 20, ...)

3. a) What are the differences between if-else block and try-catch block for exception handling? Write a program to handle divide by zero exception using appropriate try-catch block. (13)
   b) Give an example of overloading const and Non-const fùnction-with appropriate code. (08)
   c) How can you initialize a class using initialization list when all the member variables are public? (05)
   d) What is an inline function? When an inline function does not work? Explain with example. (09)

4. a) What is bit field? How is bit field used to utilize the memory space in a better way? Give an example. (10)
   b) Consider the following program- (15)

```
Class Person
{
  public:
    string name;
    int id;
    .........
    .........
}
```

Use STL map to map an integer to a set of person. So the key value pair would be:

‹key, value›≡‹int, set‹person››

Now write necessary C++ code to –
i) insert data into the map
ii) traverse the entire map

   c) How can you change the default order of entering data into priority Queue? Use custom sort object and overload "()" operator. (10)

SECTION B

(Answer **ANY THREE** questions from this section in Script B)

5. a) Discuss the possible ways of overloading Double Subscript operator "[][]" in C++. (10)

   b) How does C++ compiler differentiate between overloaded postfix and prefix decrement operators? Explain with appropriate example(s). (05)

   c) While overloading an operator using non-member function, is it mandatory to make that function a friend function? If so, explain why. (05)

   d) Consider the input.txt file shown below. Write a C++ program which reads two complex numbers from each line of input.txt file and performs the specified mathematical operation (+, –) on those numbers. For working with these numbers create a ComplexNumber class with appropriate member attributes and functions. Specified mathematical operation must be performed using overloaded + and – operators. The program must generate a file named output.txt as shown below. For printing complex numbers use overloaded << operator. (15)

```
Perform following operations:
3+2i + 5+3i
5+3i – 3+2i
```

input.txt

```
3+2i + 5+3i = 8+5i
5+3i – 3+2i = 2+1i
```

output.txt

6. a) Create an array container class with bound-checking and dynamic resizing (when elements are added or removed) capability. (15)

   b) What is memory leak? Explain with suitable example(s) how memory leaks occur in a C++ program. (08)

   c) What is the problem associated with shallow copy? Explain how the problem can be solved using deep copy (consider both copy constructor and assignment operator). (12)

7. a) Differentiate among composition, aggregation and inheritance with appropriate example. (15)

   b) Briefly explain the roles of different access specifiers used in inheritance in context of C++ language. Also discuss how granting access works for both member variables and member functions. (08)

   c) What is the problem associated with multiple inheritance? Discuss all possible ways of solving the problem. (07)

   d) "Child's pointer can point to parent's object but parent's pointer cannot point to child's object" – Justify this statement. (05)

8. a) Discuss the benefits of using "Namespace" in C++ with suitable example(s). (06)

   b) Write a C++ program to show the use of Run–Time Type Identification (RTTI) and Dynamic cast. (08)

   c) Describe the necessity of virtual destructor with an example. (05)

   d) Explain the necessity of pure virtual function with appropriate example(s). What is an Abstract class? (10)

   e) "Virtual functions are hierarchical" – justify this statement through suitable example. (06)

KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY

B.Sc. Engineering 2nd Year 1st Term Examination, 2017

Department of Computer Science and Engineering

CSE 2101

Object Oriented Programming

TIME: 3 hours FULL MARKS: 210

N.B. i) Answer **ANY THREE** questions from each section in separate scripts.
ii) Figures in the right margin indicate full marks.

SECTION A

(Answer **ANY THREE** questions from this section in Script A)

1. a) Write down minimum number of constructors to declare six objects of any user-defined class using following parameters as a constructor parameter: (08)
i) ('A', 3, 3.2) , ii) (2, 3.4) , iii) ('B', 2) , iv) (2, 3.3, 'A') , v) (2, 3.4, 2) , vi) ('C').

b) Create a STUDENT class that includes a student's first_name (private), roll_number (private) and marks of three subjects (private). Now create a demo array of objects and copy the previously created objects into this demo one using copy constructor. (12)

c) How can we prevent class objects from being copied to other object? Explain with examples. (10)

d) Explain the differences between references and pointer with example (05)

2. a) Define a class "String" that could work as a user defined string type. Include constructor that will enable you to create an uninitialized string. (15)
String S1; //sting with length 0.
And also to initialize an object with a string constant at the time of creation like.
String S2; ("object oriented");
Now, overload '+' operator to add two strings making a third string and '==' operator to check whether the two strings are equal or not. (Do not use any string related built in function).

b) Can we overload *main*() in C++? Explain your position with proper reasons. (05)

c) Why overloading of operator '<<' must return by reference? Is it possible to overload '<<' operator using member function? Explain with examples. (10)

d) Predict the output of the following program: (05)

```
#include <iostream>
using namespace std;
class Test{
        static int x;
    public:
        static int y;
        void func(int x){
          cout<<"Value of Test :: x is" <<Test::x;
          cout<<"Value of x is" <<x;
        }
    };
int Test::x = -1;
int Test::y = 2;
int main(){
      Test obj;
      int x = 3;
      obj.func(x);
      cout<<"\n Test::y = "<<Test::y;
      return 0;
}
```

3. a) What mistake prevents the following class declaration from functioning properly as an abstract class? (08)

```
class Shape{
    public:
       virtual double print()const;
       double area()const{
          return base * height;}
    };
    private:
       double base;
       double height;
};
```

b) From the following class definition find out the errors. Rewrite the program after removing errors and then predict the output of the modified program. NOTE: You cannot define any new constructor for any classes and the given constructor for each class must take only one integer value as a parameter except A1's default constructor. (15)

```
#include <iostream>
using namespace std;
class A1{
    public:
       A1(int x){
         cout<<"A1 :: A1(int) called"<<endl;}
    };
class A2 : public A1{
       A2(int x){
         cout<<"A2 :: A2(int) called"<<endl;}
    };
class A3 : public A1{
       A3(int x){
         cout<<"A3 :: A3(int) called"<<endl;}
    };
class A4 : virtual public A2{
    public:
       A4(int x){
         cout<<"A4 : A4(int) called"<<endl;}
    };
class A5 : virtual public A3{
    public:
       A5(int x){
         cout<<"A5 : A5(int) called"<<endl;}
    };
class A6 : public A4, public A5{
    public:
       A6(int x){
         cout<<"A6 :: A6(int) called"<<endl;}
    };
int main(){
    A6 A61(30);
    return 0;
}
```

c) Define two classes Polar and Rectangle to represent in the Polar and Rectangle system. Use conversion routines to convert from one system to another. (12)

4. a) Design a class hierarchy for persons (class Person), students (class Student), and master students (class MasterStudent), such that a person has a name (of type string), a student is a person with an additional matric_number (of type int), and a master student is a student with an additional subject (of type string). Provide suitable constructors for the classes and make sure that all data members are private. Write a virtual member function info that prints all information (i.e. the data members) about a person and override this function in the derived classes such that also the additional information about students and master students is printed. When overriding info, call the info function of the base class first and then print the additional information. (15)

b) What do you mean by "*this*" pointer? Explain with example. (06)

c) Why is the use of the '<<' and '>>' operators called "formatted" I/O and the use of the functions *put*(), *get*(), *write*(), *read*() etc. called "unformatted" I/O. (07)

d) Why argument to a copy constructor must be passed as a reference? (07)

SECTION B

(Answer **ANY THREE** questions from this section in Script B)

5. a) "A deep copy allocates memory for the copy and then copies the actual value, so that the copy lives in distinct memory from the source". Explain with appropriate code. (08)

b) Is it possible to use template version of Lambda Expression? Justify your opinion briefly. (06)

c) What is the difference between list's member function *remove*() and algorithm's *remove*()? (05)

d) Traverse the map: `map<string,list<map<string,vector<int>>>>` (06)

e) Write down the template version background code for the following code segment. (06)

```
for each(vect.begin(),vect.end(),print it);
```

Where "*vect*" is a vector and "*print_it*" is a function.

f) What will be the output of the following code? (04)

```
class Lsb_less{
    public:
        bool operator()(int x, int y){
            return (x%10)<(y%10);
        }
    };
set <int,Lsb_less>s = {21,23,26,27};
set <int,Lsb_less> :: iterator itrl, itr2;
itrl = find(s.begin(), s.send().36); //output = ??
itr2 = s.find(36); //output = ??
```

6. a) (08)

```
int main()
{
    ___ {                                    ___ ___
            Auto_ptr <Resource> res (new Resource)
    }
    return 0;
}
```

Use smart pointer concept and append necessary code with the above code segment in such a way that "*res*" gets deallocated when it goes out of scope.

b) What would be the problem when you write the following code segment? (*Auto_ptr* is a smart pointer). (05)

```
Auto_ptr <Resource> res1 (new Resource);
Auto_ptr <Resource> res2 (new res1);
```

c) Complete ~~Compute~~ the following two tables. (06)

L-value reference	Can be initialized with	Can modify
Modifiable L-values	—	—
Non-Modifiable L-values	—	—
R-values	—	—

R-value reference	Can be initialized with	Can modify
Modifiable L-values	—	—
Non-Modifiable L-values	—	—
R-values	—	—

d) (08)

```
class Dog{
    public:
        void bark(){
            cout<<"I don't have a name."<<endl;
        }
    };
class YellowDog : public Dog{
        string m_name;
    public:
        YellowDog(string name):m_name(name){}
        void bark(){cout<<"My name is "<<m_name<<endl;}
};
```

Write necessary codes (use STL containers) to arise a scenario of "object slicing" and then resolve it.

e) (08)

```
template <class T>
class storage8{
private:
T_m_array[8];
public:
void set(int index, const T &value)
  { m_aray[index] = value;}
  ----------------
};
```

```
int main(){
 storage8<int>intstorage;
 for(int c = 0;c < 8;c++)
      intstorage.set(c,c);
 storage8<bool>bool.storage
 for(int c = 0;c < 8;c++)
      boolstorage.set(c,c&3);
 return 0;
}
```

*storage*8<*bool*> class in the above code segment contains 8 bools, which is 1 byte worth of useful information and 7 byte of wasted space. Now, using the concept of class template initialization, append necessary code with the above code segment so that the *array* of 8 bools can be replaces with a variable that is single byte in size.

7. a) Give two solutions for the problem in the following code segment (04)

```
try{
        Person *pJohn = new Person("John", 18);
        ProcessPerson(pJohn);//Exception occurs here
        delete pJohn;
    }
catch(PersonException &CE)
    {
        cerr<<"Fail: "<<&CE.what();
    }
```

b) Give an example of ~~stock~~ stack undwinding via exception handling. (07)

c) "Exceptions should not be thrown in destructors". Why? (05)

d) What does it mean to delete memory and what is dangling pointer? (06)

e) What are the problems of writing pointers to disk? (07)

f) Identify the problem in each of the following codes. (06)

```
i) void doSomething()
   {
       int *ptr = new int;
   }
```

```
ii) int value = 5;
    int *ptr = new int;
    ptr = &value;
```

```
iii) int *ptr = new int;
     ptr = new int;
```

8. a) Write necessary code to read the content from an input file (in.txt) and make an output file that looks like out.txt. Use file pointer manipulator concept for this. (06)

```
This is line 1
This is line 2
This is line 3
This is line 4
```

in.txt

```
is line 1
line 2
his is line 4
```

out.txt

b) What are the errors in the following code segments? (if any) (04)

`vector<int>vec = {2,3,4,5};`
`int *p = &vec[3];`
`vec.push_back(6);`
`cout<<*p<<endl;`
	`deque<int>deq = {2,3,4,5};`
	`p = &deq[3];`
	`deq.push_back(6);`
	`cout<<*p<<endl;`

c) Given a key, use Lambda Expression and find_if algorithm to find the corresponding value in an STL map? (08)

d) What are the ways to insert a custom object in a set? Explain with appropriate code only. (08)

e) What is anonymous variable? Is it possible to hold the address of an anonymous object? Justify your answer. (05)

f) What is meant by each of the following code segments: (04)

```
i)    char *const ptr1 = "ABC";
ii)   int const *ptr2 = &m;
iii)  const char* const cp = "XYZ";
iv)   const static int x = 42;
```

KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY
B.Sc. Engineering 2nd year 1st term Examination, 2018
Department of Computer Science and Engineering
CSE 2101
Object Oriented Programming

TIME: 3 hours FULL MARKS: 210

N.B. i) Answer **ANY THREE** questions from each section in separate scripts.
ii) Figures in the right margin indicate full marks.

SECTION A

(Answer **ANY THREE** questions from this section in Script A)

1. a) Distinguish between the following terms: (08)
   i) Objects and classes,
   ii) Data abstraction and data encapsulation.
   b) "The speed benefits of inline function diminish as the function grows in size." -- justify the statement. (07)
   c) Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a C++ program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object. The output should be in the format of feet and inches or meters and centimeters. Assume 1 centimeter is equal to 0.39 inch. (12)
   d) Give an example of overloading constant and Non-constant function with appropriate examples. (08)

2. a) What is inline function? When an inline function does not work? Explain with example. (08)
   b) Is it possible to call constructor and destructor explicitly? Explain with suitable example. Also explain the output. (08)
   c) Write a C++ program to create a class "DoubleArray". It will be used to store double data in an array member variable. The data will be stored in heap. At first create an object and store N double data. Later, find the average of the data and copy the average and data into another object during initialization of that object using deep copy method. (12)
   d) "Static member function has no this pointer." – Is it true? Explain your answer with example. (07)

3. a) What are the advantages of using this pointer? Explain with suitable example(s). (09)
   b) Write a C++ program to add two 2×2 Matrix of complex numbers. Write the declaration of complex number class into ComplexNumber.h file and definition into ComplexNumber.cpp file. Then create a template class MATRIX to perform matrix addition. (14)
   c) What is STL? Briefly describe the components of STL. Also, give a C++ coding example. (12)

4. a) Explain the uses of functor with appropriate example(s). What are the advantages of functor over function pointer? (09)
   b) Define Lambda expressions. Explain the syntax of Lambda expression with suitable example(s) (08)
   c) Compare the performance characteristics of lists, vectors and maps with proper example. (11)
   d) Define reference. What are the typical uses of reference? (07)

SECTION B

(Answer **ANY THREE** questions from this section in Script B)

5. a) What do you mean by containership? How does it differ from inheritance? (08)
   b) Briefly describe the multiple and hierarchical inheritance with examples. (08)
   c) Where does the diamond problem occur in inheritance? How does it can be solved? Give example. (13)

d) Explain the output of the following program. (06)

```
#include<iostream>
using namespace std;
class A{
  public:
    A(int a){
     cout<<”Base A \n”;
  }
};
class B{
  public:
    B(int a){
     cout<<”Base B \n”;
  }
};
class C: protected A, private B{
  public:
    C(int a):B(a),A(a){
     cout<<”Derived C \n”;
  }
};
int main(){
   C c(10);
   B a(5);
}
```

6. a) Define polymorphism, static binding and dynamic binding. (06)
   b) Consider the following code segment. (08)

```
class M{
  public:
    int a, b;
  private:
    double d;
  public:
    void set(int a)
     {
          //...
     }
 };
int main()
{
     M obj1;
     M obj2;
}
```

Discuss the amount of memory allocated by obj1 and obj2.
   c) Why is pure virtual function necessary? Explain with proper example. (10)
   d) How do we achieve runtime polymorphism? Explain with proper example. (11)

7. a) How does exception differ from error? (05)
   b) Describe the exception handling mechanism using block diagram. (10)
   c) How does rethrowing of exception work? Give an example. (10)
   d) How does C++ handle different signals? What is the general syntax of signal() function? Explain. (10)

8. a) Why is Run Time Type Identification (RTTI) a necessary feature of C++? (05)
   b) What are the design principle of Object Oriented Programming? (08)
   c) Explain Dynamic Cast using an example base class and derived class. (10)
   d) Briefly describe the different membership function of type_info class for RTTI. (12)

KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY

B.Sc. Engineering 2nd Year 1st Term Examination, 2019

Department of Computer Science and Engineering

CSE 2101

Object Oriented Programming

TIME: 3 hours FULL MARKS: 210

N.B. i) Answer **ANY THREE** questions from each section in separate scripts.
ii) Figures in the right margin indicate full marks.

SECTION A

(Answer **ANY THREE** questions from this section in Script A)

1. a) Discuss about Data Hiding with necessary figure. What are the limitations of C structure? (10)
   b) What are the special characteristics of a static member variable? How can you access static variable and functions of a class without creating any object? Explain with example code. (10)
   c) Define friend function. Write the special characteristics of friend function. (10)
   d) Explain the difference between references and pointer with example. (05)

2. a) "When default argument constructor is called with no argument, it becomes a default constructor."-justify the statement with proper example. (10)
   b) Is it possible to call constructor and destructor explicitly? Explain with suitable example. Also explain the output. (10)
   c) How constructors are different from a normal member functions? What are the advantages of *this* pointer? (10)
   d) Why should the copy constructor accept its parameter by reference in C++? (05)

3. a) How does C++ compiler differentiate between overloaded postfix and prefix decrement operators? Explain with appropriate examples. (10)
   b) Write a C++ program to add two 2X2 Matrix of complex numbers. Write the declaration of complex number class into ComplexNumber.h file and definition into ComplexNumber.cpp file. Then create a template class Matrix to perform matrix addition. (12)
   c) Is it possible to overload "<<" operator using member function? Explain with example. (08)
   d) Discuss memory allocation of objects in Object-oriented Programming (OOP). (05)

4. a) What is STL? Discuss about the types of STL containers. Give an example how does stack and priority queue work? (12)
   b) Write a program to add two numbers and store the result using file. Sample input and output are given below. (10)

10 20	30
30 40	70
50 10	60

input.txt output.txt

   c) Why do we use template? Predict and explain the output of the following code: (08)

```
template <class T>
void fun(const T &x)
{
    static int count = 0;
    cout << "x = " << x << " count = "<< count++ <<endl;
    return;
}
int main()
{
    fun<int> (1);
    fun<int> (2);
    fun<double> (1.1);
    return 0;
}
```

   d) How can you initialize a class using initialization list when all the member variables are public? (05)

SECTION B

(Answer **ANY THREE** questions from this section in Script B)

5. a) Write down the differences between Structured Programming Language and Object-oriented Programming Language. (10)

   b) What is the problem arise in the multiple inheritance? How can you overcome ti? (09)

   c) Why do we need "protected" access modifier? (06)

   d) Explain the output of the following program: (10)

```
#include <iostream>
using namespace std;
class A {
    public:
      A(){cout<<"_1 ";}
    A(A & obj){ cout<<" 2 ";}
};
class B: virtual A{
    public:
      B(){cout<<" 3 ";}
    B(B & obj){ cout<<" 4 ";}
};
```

```
class C: virtual A{
    public:
      C(){cout<<" 5 ";}
    C(C & obj){ cout<<" 6 ";}
};
class D: B, C{
    public:
      D(){cout<<" 7 ";}
    D(D & obj){ cout<<" 8 ";}
};
int main(){
    D d1;
    D d (d1);
}
```

6. a) "Every interface is abstract class but every abstract class is not interface"-justify the statement with examples. (10)

   b) Why is pure virtual function necessary? Explain with proper example. (10)

   c) Explain the compile time polymorphism and run time polymorphism in details with examples. (15)

7. a) Why is Run Time Type Identification (RTTI) a necessary feature of C++? (05)

   b) Design a class containing attributes *x* and *y* where (*x*,*y*) denotes a 2D point. Derive another class which contains a value *z* and reuse the values *x* and *y*. So, the objects of derived class denotes a 3D point. Use constructor in both classes to initialize an object of derived class. Then declare and initialize two objects (3D points) of derived class and calculate the distance between these points. Use membership functions as you like. Write the entire code in C++. (15)

   c) Explain Dynamic cast using an example of base class and derived class. (10)

   d) How does exception differ from error? (05)

8. a) Why do we need to handle exceptions in program? (05)

   b) What do you mean by Is-A relation and Has-A relation? Discuss with examples. (10)

   c) Is there exist any abnormality in the following code? If yes then how can it be handled? (10)

```
#include <iostream>
using namespace std;
int main(){
    int array[10] = {10};
    for (int i = 0; i <= 10; i++)
        cout << array[i] << endl;
}
```

   d) How can we restrict a function to throw certain exception? (10)

KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY
B.Sc. Engineering 1st Year 2nd Term Examination, 2022
Department of Computer Science and Engineering
CSE 1205
Object Oriented Programming

TIME: 3 hours FULL MARKS: 210

N.B. i) Answer **ANY THREE** questions from each section in separate scripts.
ii) Figures in the immediate right column of the questions indicate full marks.
iii) The rightmost column indicates course outcomes.

SECTION A

(Answer **ANY THREE** questions from this section in Script A)

1. a) Differentiate among composition, aggregation and inheritance with appropriate example(s). (12) [CO1]

b) What is the problem of shallow copy? Provide a solution to this problem with coding example. (10) [CO2]

c) Design two classes, Polar and Rectangle to represent points in the polar and rectangle systems. Use constructors to convert from one system to another. Now, write down the full program showing the conversion from Polar point to Rectangle point and vice-versa. (13) [CO3]

2. a) When is virtual destructor needed? Write down the scenario with programing example. (07) [CO2]

b) Predict the output of the following program, if any error occurs, then provide the solution by redefining the class(es) only, without changing main(). (10) [CO3]

```
class A{
  public:
  virtual void fun()=0;
  };
```

```
class B:public A{
  public:
      void fun(){
      cout<<"B fun"<<endl;
         }};
```

```
class C: public B{
     };

class D: public A{
     };
```

```
int main()
{   A *P; B b; C c; D d;
    P=&c;
    P→fun();
    P=&d;
    P→fun();
    return 0;}
```

c) Explain assertion in C++ with coding example. (06) [CO1]

d) Design a C++ program to add two 2×2 matrix of complex numbers. Write the declarations of complex number class into *ComplexNumber.h* file and definition into *ComplexNumber.cpp* file. Then create a template class Matrix to perform matrix addition. (12) [CO4]

3. a) Explain the constructor calling order in inheritance by listing all the steps and using appropriate example. (10) [CO3]

b) Will the program shown in the following figure compile successfully? If not, explain what modification should be done to compile it successfully. (12) [CO2]

```
#include<iostream>
template<typename T>
T max(T a, T b){
if(a>b) return a;
else return b;
}

class Test{
private: int val;
public:
        Test(int val():val(val)
         {} }
```

```
int main()
{
  Test t1=10, t2;
  t2=5;
  Test t3=max(t1, t2);
  std::cout<<t3;
  return 0;
}
```

c) How exception can be handled in C++? Explain with appropriate example. (06) [CO4]

d) Explain "Has-A-Relation" with proper example. (07) [CO2]

4. a) How does virtual functions in C++ work? Explain with proper illustration. (06) [CO3]

b) Write a program that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space. (11) [CO1]

c) Create a class "Polygon" with two pure virtual functions, "area()" to calculate area and "name()" to display class name. Now, inherit this class to three other classes with following member variables: Rectangle(height, width), Triangle(height, width), Square(side). (18) [CO4]

(i) Create a function showDetails() with Polygon *parameter. Print a line "It's a Rectangle" if it is Rectangle object, do similar for other classes and show the area().

(ii) Create a function showDetails Using RTTI() to produce the similar result of (i) but using RTTI and dynamic casting instead of using name() function.

SECTION B

(Answer **ANY THREE** questions from this section in Script B)

5. a) What are the key differences and characteristics that set a C++ structure apart from a C++ class? Describe briefly. (08) [CO1]

b) Create two classes, TimeHM and TimeMS, which represent time values. The TimeHM class stores time in hours and minutes, while the TimeMS class stores time in minutes and seconds. Design a program that reads values for objects of these classes and adds one TimeHM object to another TimeMS object. Use a friend function to perform the addition operation. (15) [CO2]

c) Clarify how std::map handles duplicate keys, and when multiple key-value pairs with the same key are allowed. Provide an example illustrating the use of std::multimap for string duplicates. (12) [CO4]

6. a) In the realm of C++, the allocation of memory dynamically can be accomplished using both 'new' and 'malloc' methods, which one do you prefer, and why? Justify your choice with an example. (12) [CO2]

b) Imagine you are on a team building software for a big university's library system. The system has multiple components, including a cataloging module having Book class, a borrowing and returning module having Library class, and a notification module having Notification class. These components need to share essential information and functionality. Explain how you would use static number variables within the classes representing these modules to maintain shared data. (15) [CO1]

c) Define a 'Namespace conflict' in C++ and provide a scenario where it can occur. Explain in detail how a 'Namespace conflict' can be resolved in C++? (08) [CO4]

7. a) Describe the concept of custom signal handlers and their importance in application development. Demonstrates how to set up a custom signal handler for the SIGINT signal and specify what actions it should perform when the signal is received. (10) [CO1]

b) Mr. X is developing a software application to manage different types of products in a store. To achieve this, he decides to create a generic class named 'Product' with two member variables 'product_id ' which is an integer by default but can vary, and 'product_price' which can be a floating point value representing the price. Now, design the generic 'product' class, including both a parameterized and a default constructor, to handle a wide range of product types and their respective prices. (15) [CO3]

c) What problem arises in the following code scenario? How can you solve this problem? Explain with proper code. (10) [CO2]

```
class Data {
   int *data;
     };

void process()
 {
  Data *pData=new Data();
 }
```

```
int main()
{
 while(1)
  {
    process();
  }
}
```

8. a) "The speed benefits of using inline functions diminish as the function grows in size". Justify this statement by discussing the relationship between function size, inline function optimization. (10) [CO1]

b) Consider the following program. Apply operator overloading using friend function to overload required operators of Product class, such that program executes without any error. (13) [CO3]

```
class Product{
 Public:
   int product_id;
   string product_name;
   double product_price;
};
```

```
int main()
{
 ifstream in("input.txt")
 ofstream out("output.txt")
 Product P;
 in>>P;
 out<<P;
}
```

c) What would be the problem when you write the following code segment. (12) [CO4]

```
unique_ptr<MyClass> myPtr(new MyClass());
unique_ptr<MyClass> myPtr2=myPtr
```

How can you fix the problem using std::unique_ptr, and are there any other smart pointers that could be used to solve this issue ? if so, explain how those alternative smart pointers can be utilized to resolve the problem.