

**National University**

**of Computer & Emerging Sciences Peshawar Campus**



Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examination: Sessional-II

Total Marks: **30** Weightage: **15**

Date: 7th April, 2023

Instructor: Sara Rehmat

Program: BS(SE), BS(AI)

Semester: Spring-2023

Time Allowed: 1:00 hour

Course: Object Oriented Programming (CS1004)

**NOTE:** Attempt all questions.

|  |  |  |
| --- | --- | --- |
| **Question 1: CLO 1** | **Marks: 2\*5 = 10** | **Expected Time: 15 minutes** |

Give short answers to the following questions:

1. Describe two situations in which the copy constructor executes.
2. Can an object of one class be assigned to an object of another class? Justify your answer.
3. What is the purpose of including #ifndef statement in your program?
4. Differentiate between method overloading and method overriding in inheritance.
5. Write the order of the execution of the constructors and destructors in inheritance.

|  |  |  |
| --- | --- | --- |
| **Question 2: CLO 2** | **Marks: 6+4= 10** | **Expected Time: 20 minutes** |

Consider the following program with the assumption that necessary header files and namespaces have been included. The code contains some syntactical errors.

1. Rewrite the code after correcting the errors. The errors must be highlighted in the form of comments in the corrected code.
2. Write the output of the corrected code.

|  |  |  |
| --- | --- | --- |
| class A  {  private:  int x;  protected:  int y;  public:  A()  {  x=0;  y=0;  }  virtual void print() const = 0;  }; | class B: protected A  {  protected:  int z;  public:  B(int a=0)  {  z = a;  }  void print() const{  cout << y << " " << z <<endl; }  }; | class C: public A  {  private:  int w;  public:  void print() const{  cout << y <<" "<< x << " " << w << endl;}  };  class D:public B, private C  {}; |

|  |  |  |
| --- | --- | --- |
| int main()  {  A a;  B b(7);  D d;  d.print();  b.print();  A \*a1 = new C;  a1->print();  } | | |
| **Question 3: CLO 3** | **Marks: 10** | **Expected Time: 20 minutes** | |

Playing cards are divided into four suits: spades, hearts, diamonds, and clubs. A deck of cards includes 13 cards in each suit (10 numbered cards and 3 court cards—king, queen, and jack).

Create a program to shuffle and deal a deck of cards. The program must consist of class **Card**, class **DeckOfCards** and a test program.

Class Card should provide:

a) Data members *face* and *suit* of type **int**.

b) A constructor that receives two integers representing the face and suit and uses them to initialize

the data members.

c) A setter to set the values for face and suit.

d) Two static arrays of strings representing the faces and suits.

e) A *toString* method that returns the Card as a string in the form “face of suit” e.g. six of hearts. You can use the + operator to concatenate strings.

Class **DeckOfCards** should contain:

a) An array named *deck* to store the objects of class **Card**.

b) An integer *nextCard* to represent the index containing the next card to deal. Initially it will be zero.

c) A default constructor that initializes the array *deck* and then sets all the cards. There are 52 cards in a deck.

d) A *shuffle* method that shuffles the cards in the deck. The shuffle algorithm should iterate through the array of cards. For each Card, randomly select another Card in the deck and swap the two Cards. For selecting a random integer, use the *rand()* function.

e) A *dealCard* method that returns the next Card object from the deck. The next Card is the card at the index *nextCard*.

f) A *moreCards* method that returns a bool value indicating whether there are more Cards to deal.

The test program should create a **DeckOfCards** object, shuffle the cards, then deal the 52 cards.