Explanations:

1. **[4 Marks]** What is function overloading? Give two examples. To which aspect of object oriented programming does it belong?
2. **[4 Marks]** Explain the keywords public, protected and private? To which aspect of object oriented programming do they belong?
3. **[3 Marks]** Explain shallow and deep copying in classes. Identify the type of the class that is suitable for each.
4. **[4 Marks]** Ad-Hoc polymorphism is divided into two classifications. Name and explain each classification with one example.
5. **[2 Marks]** What is a pure virtual function? What does the presence of a pure virtual function do to a class?
6. **[2 Marks]** What is a friend function? Explain its advantages or disadvantages.

1. **[2 Marks]** What is a safe empty state in a class? Which member function usually sets this state?

1. **[2 Marks]** What is the output of the following source code?

class Val {

int m\_val;

public:

Val(int v) { m\_val = v; }

Val(const Val& V) {m\_val = V.m\_val - 1; }

void disp()const { cout << m\_val << endl; }

};

Val foo(Val V) { return V; }

int main() {

Val A(10);

Val B = foo(A);

B.disp();

}

1. **[2 Marks]** Given the following class:

class Name {

char m\_value[100];

public:

Name(const char value[] = "");

Name& operator=(const char value[]);

Name& operator=(const Name& N);

};

Which member function does the following code snippet invoke? Why?

Name N = "Homer";

**Walkthrough:**

**Question 6 (A) - (6 marks)** Determine the exact output of the following program:

#include <iostream>

using namespace std;

class Calc

{

char Grade;

int Bonus;

public:

Calc()

{

Grade = 'E';

Bonus = 0;

}

void Down(int G)

{

Grade -= G;

}

void Up(int G)

{

Grade += G;

Bonus++;

}

void Show()

{

cout << Grade << "#" << Bonus << endl;

}

};

int main()

{

Calc C;

C.Show();

C.Down(1);

C.Show();

C.Up(2);

C.Show();

C.Down(5);

C.Show();

}

**Question 6 (B) - (6 marks)** Determine the exact output of the following program:

// WalkA.cpp

#include <iostream>

using namespace std;

class playdoh {

int weight;

public:

playdoh(int w = 5){

weight = w;

cout << "Doh: " << weight << endl;

}

void add(){

weight += 6;

cout << "Play: " << weight << endl;

}

~playdoh(){

weight -= 4;

cout << "Clay: " << weight << endl;

}

};

void mixup(playdoh doh1, playdoh& doh2){

doh1.add();

doh2.add();

}

int main(){

playdoh p1(15);

playdoh p2;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

mixup(p1, p2);

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

mixup(p2, p1);

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

return 0;

}

**Question 6 (C) - (16 marks)** Determine the output of the following program:

#include <iostream>

using namespace std;

class Wind {

int category;

public:

Wind(int cat = 3) {

category = cat;

cout << "1." << cat << endl;

}

virtual void increase(int amount) {

category += amount;

cout << "A. " << category << endl;

}

void operator++() {

++category;

cout << "B. " << category << endl;

}

virtual ~Wind() {

cout << "C. " << category << endl;

}

};

class Tornado : public Wind {

double velocity;

public:

Tornado(int cat, double vel) : Wind(cat) {

velocity = vel;

cout << "2. " << vel << endl;

}

virtual void increase(int value) {

velocity += value;

cout << "X. " << velocity << endl;

}

void operator++() {

Wind::operator++();

velocity += 20;

cout << "Y. " << endl;

}

~Tornado() {

cout << "Z. " << velocity << endl;

}

};

int main() {

Wind\* wind\_array[2];

wind\_array[0] = new Tornado(7, 66.5);

wind\_array[1] = new Wind(5);

for (int i = 0; i < 2; i++) {

wind\_array[i]->increase(5);

++(\*wind\_array[i]);

}

for (int i = 0; i < 2; i++)

delete wind\_array[i];

return 0;

}

**Code Completion:**

**Question 7 – (5 Marks)** Define a C++ template (generic) function **Large** (single definition should work for different data types, e.g., int, float, char, …….) which accepts two parameters of the same type and has a return type. The function should return the largest among the two parameters using function templates.

**Question 8 – (5 Marks)** Consider the following incomplete class definitions. Fill in the missing code as described in the comments. Your solution should call the appropriate set and get functions on a Book object.

class Book{

public:

//Provide:

// - a pure virtual function to set the ISBN/eISBN

// - a pure virtual function to get the ISBN/eISBN

};

// create an EBook class that inherits from Book

{

int eISBN;

public:

void set(const int \_eISBN){eISBN=\_eISBN;};

int get() const{return eISBN;};

};

// create a PaperBook class that inherits from Book

{

int isbn;

public:

void set(const int \_isbn){isbn=\_isbn};

int get() const{return isbn;};

};

**Code Writing:**

**Question 9 – (10 Marks)** Declare a class template Calculator. The class contains

1. Two private members num1 and num2 of user defined data types
2. A constructor to initialize the members.
3. Two public member functions multiply and divide to calculate multiplication and division of the numbers and returns the value of data type defined by the user.
4. A function displayResult() to display the final output to the screen.

**The program should match the following output:**

Int Results:

Numbers Are: 2 and 1

Product is: 2

Division is: 2

Float Results:

Numbers are: 2.4 and 1.2

Product is: 2.88

Division is: 2

1. **Question 10 -** This question has 2 parts: a) code an abstract base class and b) derive a class from your abstract base class.

a) **[6 Marks]** Create an abstract base class named **Abstract\_Employee**. This class has no data members and includes two pure virtual functions: a modifier named **set\_data()** and a query named **display\_data()**. This class will be the base class for your answer to the next question.

b) **[16 Marks]** Derive a class named **Teacher** from your **Abstract\_Employee** class, with the following properties/member functions:

* A char array that holds the name (the length is unknown). [1 mark]
* An integer that holds an employee identifier. [1 mark]
* A public constructor that takes two arguments, an integer and the address of an unmodifiable char array. Using these arguments your constructor initializes the values of the teacher’s name and identifier. Ensure that your constructor does not corrupt memory. [3 marks]
* A public **set\_data()** member function. This function obtains the identifier using the **set\_data()** function of the base classand then asksthe user for the teacher’s name. For this function only, you may assume that the user will only enter up to 100 characters. [3 marks]
* A public **display\_data()** query that displays the teacher’s name and identifier. [2 marks]
* Make sure a **Teacher** can be copied but cannot assigned to another **Teacher**. [3 mark]
* Make sure there is no memory leak after a **Teacher** object goes out of scope. [2 marks]