

Key

OOP-FALL22;QUIZ-01;DATE:14-11-2022; TIME:30MIN; MARKS:50

NAME: \_\_\_\_\_ ID: \_\_\_\_\_

Q.1: Mark the following statements as true or false. (2\*5=10marks)

- a. The member variables of a class must be of the same type. **F**
- b. The member functions of a class must be public. **F**
- c. A class can have more than one constructor. **T**
- d. A class can have more than one destructor. **F**
- e. Both constructors and destructors can have parameters. **F**

Q.2: Consider following class definition. Answer following questions. (4\*10=40marks)

Consider the definition of the following class:

```
class employee //Line 1
{ //Line 2
public: //Line 3
    employee(); //Line 4
    employee(string, int, double); //Line 5
    employee(int, double); //Line 6
    employee(string); //Line 7

    void setData(string, int, double); //Line 8
    void print() const; //Line 9
    void updatePay(double x); //Line 10
    int getNumOfServiceYears() const; //Line 11
    double getPay() const; //Line 12

private: //Line 13
    string name; //Line 14
    int numOfServiceYears; //Line 15
    double pay; //Line 16
}; //Line 17
```

- a. Give the line number containing the constructor that is executed in each of the following declarations:
  - a. employee tempEmployee; **Line 4**
  - b. employee newEmployee("Harry Miller", 0, 25000); **Lines 5**
  - c. employee oldEmployee("Bill Dunbar", 15, 55000); **Lines 6**
- b. Write the definition of the constructor in Line 4 so that the instance variables are initialized to "", 0, and 0.0, respectively.

```
employee() {
    name = "";
    num of Service Years = 0;
    pay = 0.0;
}
```

- c. Write the definition of the constructor in Line 5 so that the instance variables are initialized according to the parameters.

```
employee(string n, int y, double P) {
    strcpy(name, n);
    No of Service Years = y;
    pay = P;
}
```

- d. Write the definition of the constructor in Line 6 so that the instance variable name is initialized to the empty string and the remaining instance variables are initialized according to the parameters.

```
employee(int y, double P) {
    name = "";
    No of Service Years = y;
    pay = P;
}
```

- e. Write the definition of the function setData so that the instance variables are set according to the parameters.

```
void setData (String n, int Y, double P) {  
    name = n;  
    NoofServiceYears = Y;  
    pay = P;  
}
```

- f. Write the definition of the function print to output the values of the instance variables.

```
void Print() {  
    cout << name << NoofServiceYears << pay;  
}
```

- g. Write the definition of the function updatePay to update the value of the instance variable pay by adding the value of the parameter.

```
void updatePay (double x)  
{  
    pay = x;  
}
```

- h. Write the definition of the function getNumOfServiceYears to return the value of the instance variable numOfServiceYears.

```
int getNumOfServiceYears ( ) const  
{  
    return getNoofServiceYears;  
}
```

- i. Write the definition of the function getPay to return the value of the instance variable pay.

```
int getPay ( ) const  
{  
    return pay;  
}
```

- j. Write a program to test the class employee.

```
int main()  
{  
    employee e1, e2("a", 20, 20000);  
    e1.SetData();  
    e2.Print();  
    return 0;  
}
```