JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -1 EXAMINATIONS-2022

B.Tech. -III Semester (CS/IT)

COURSE CODE (CREDITS): 18B11CI311 (3) MAX. MARKS: 15

COURSE NAME: Object-Oriented Systems and Programming

COURSE INSTRUCTORS:A.Kumar, D.Gupta, H.Jindal&S.Singh MAX. TIME: 1 Hour

Note: All questions are compulsory. Marks and COs are indicated against each question in square brackets.

Q1. Write a C++ program to create a class *CStudent* with the following specifications: [3] CO1

- An instance variable named as *score* to hold a student's 5 exam scores.
- A *void GetScore ()* function that reads 5 integers and saves them to scores.
- An *intTotalScore ()* function that returns the sum of the student's scores.

Q2. Inline function brings many advantages, however, it is simply a request to the compiler. In the same context, briefly explain (two points each) the following: CO2

a) Under what circumstances, a compiler can ignore such a request? [1.5]

b) Why the use of inline functions is always recommended instead of macros? [1.5]

Q3. Write a program in C++ to create a class consisting of a member function – *CountObjects ()*, which counts number of objects in a program, and invoke this member function without using an object. [3] CO1 CO2

Q4. Which of the following overloaded functions are allowed or not allowed in C++? Please provide a brief explanation (2-3 sentences) to justify your answer. CO1

a) float fAddition (float iLength, float iBreadth); [1]
static float fAddition (float iLength, float iBreadth);

b) float fAddition (float iLength, float iBreadth); [1]
float fAddition (float iLength, float iBreadth = 20.0);

c) float fAddition (float *pLength, float iBreadth); [1]
static float fAddition (float iLength [], float iBreadth);

Q5. What is the output (if any) of the following program? Please give brief explanation (2-3 sentences) in support of your answer. [1] [1] [1] CO1 CO2

a)

```
#include<iostream>
using namespace std;

class CTest1
{
private:
intiCount;

public:
  CTest1 ()
  {
    iCount = 0;
    cout<< "Constructor\n";
  }
  ~CTest1 ()
  {
    cout<< "Destructor\n";
  }
};

int main()
{
  intiCount = 0;
  if (iCount == 0)
  {
    static CTest1 obj;
  }
  cout<< "End of Main\n";
  return 0;
}
```

b)

```
#include<iostream>
using namespace std;

class CTest2
{
  private:
    intiCount;
    constintiValue = 10;

public:
    CTest2 (int x)
    {
       iCount = x;
    }

    void Display () const
   {
     iCount += 5;
     cout<<iCount;
     cout<<iValue;
   }
};


int main()
{
   CTest2 obj (5);
   obj.Display ();
   return 0;
}
```

c)

```
#include<iostream>
using namespace std;

class CTest3
{
  private:
intiCount;

  public:
    CTest3 ()
    {
iCount = 1;
    }

CTest3 (CTest3&obj)
{
iCount = obj.iCount;
iCount++;
cout<<iCount;
}

void fun (CTest3obj)
{
int temp = obj.iCount;
}


 };

int main ()
  {
     CTest3 obj1;
     CTest3 obj2 = obj1;
     obj2.fun (obj1);
     return 0;
  }
```