BACHELOR OF SCIENCE EXAMINATION, 2017

(Second Year, First Semester)

COMPUTER SCIENCE (SUBSIDIARY)

Paper : Unit-8

(Introduction to object oriented programming using C++)

Time : 2 hours | Full Marks : 50

Attempt Question No. 1 and any ***three*** from the rest.

1. (a) Find errors with reasons in the following functions:

```
void f1(char* p) {
    int a = 1;
    const int c = 2;
    const int* p1 = &c;
    const int* p2 = &a;
    int* p3 = &c;
    *p3 = 7;
    const int x;
}
void f2(char* p) {
    double& dr = 1;
    const double& cdr = 1;
    int i = 1;
    int& r1 = i;
    int& r2;
    extern int& r3;
}
```

[*Turn over*]

```
void f3(char* p) {
    char s[ ] = "Jadavpur University";
    const char* pc = s;
    pc[3] = 'g';
    pc = p;
    char *const cp = s;
    cp[3] = 'a';
    cp = p;
    const char *const cpc = s;
    cpc[3] = 'a';
    cpc = p;
}
```

(b) Explain: argument passing by *value* and *reference*. 5+6 = 11

2. (a) What do you mean by *declaration*, *definition*, *initialization* and *assignment* to an *object*? Explain with examples.

(b) Write the outputs when the function $f()$ is called, where

```
int x = 11;
void f() {
    int y = x;
    x = 22;
    int x = 29;
    {   int x = 2;
        cout << x << "\t" << ::x << "\t" << y << "\n";
    }
    x = 3;
    cout << x << "\t" << ::x << "\t" << y << "\n";
}
```

[*Turn over*]

(c) Distinguish *static variable* and *static member*? Explain with examples. $4 + 4 + 5 = 13$

3. Design a class to represent a complex number with two *private* data members for the *real* and *imaginary* parts and *friend operator+*: no member function or operator. Overload the operator + so that the expressions $10.2 + z$, $z + w$ and $z + 10$ can be evaluated, where z and w are complex numbers. $5 + 8 = 13$

4. (a) What are *garbage* and *danyling pointer*?

(b) Identify and explain the problems due to garbage and dangling pointer with the code:

```
class Name { const char* s; };
class Table {
    Name* p; long sz;
public:
    Table(long s = 15){ p = new Name[sz = s];}
    ~Table( ){delete[ ] p;}
};
void f( ){
    Table t1, t2 = t1, t3;
    t3 = t2;
}
```

Using *copy constructor* and *copy assignment*, modify the code to resolve the problems. $5 + 8 = 13$

5. (a) Consider the following code segment:

```
struct Employee {int d, m, y, id;};
struct Manager: public Employee {int level, group;};
```

```
void main(manager mm, Employee ee) {
    Employee* pe = & mm ;
    Manager* pm = &ee:
    pm-> level = 2:
}
```

Identify errors with explanations for the above code.

(b) Define a class as follows:

```
struct base {virtual void iam( ) {cout<<"base"\n":} }:
```

Derive two classes from *base*, and for each define *iam( )* to write out the name of the class. Create objects of these classes and call *iam( )* for them. Assign pointers to objects of the derived classes to *base** pointers and call *iam( )* through those pointers. Identify and explain the use of *polymorphism* in the above implementation. $5 + 8 = 13$

6. What do you mean by *pure virtual* function, and *abstract class* in *C++*. Explain the errors in the following code segment:

```
class Shape {
    virtual void rotate(int) = 0:
    virtual void draw(int) = 0:
    virtual void isClosed() = 0:
}:
struct Polygon: public Shape { isClosed() { return true:} }:
void main() {
    Polygon b:
}
```

$(4 + 1) + 5 = 13$