RAJARATA UNIVERSITY OF SRI LANKA
FACULTY OF APPLIED SCIENCES

B.Sc. (General) Degree in Information and Communication Technology
First Year - Semester II Examination –February/ March 2019

ICT 1306 – OBJECT ORIENTED PROGRAMMING

Time: Three (03) hours

INSTRUCTIONS TO CANDIDATES

- This paper consists of **seven (07)** pages including this page.
- This paper consists of **Five (05)** questions. Answer **ALL** questions.
- This examination accounts for 40% of the course assessment. The total maximum mark attainable is 100. The marks assigned for each question and section, thereof are indicated in brackets.
- This is a closed book examination.
- Mobile phones or any other communication devices are not permitted.

1. Examine the source code given below and answer.

```
#include <iostream>
#include <string>
class Salary
{
  private:
        double basic;
         double overtime;
        double fullSalary;
        void calculateOvertime(int hrs, double payperhr)
         {
                overtime = hrs*payperhr;
        }
         void calculateFullSalary()
        {
                fullSalary = basic+overtime;
        }
  public:
        Salary(double bsic):basic(bsic),overtime(0.0),fullSalary(0.0)
         {}

         double getFullSalary(int hrs, double payperhr)
         {
                calculateOvertime(hrs, payperhr);
                 calculateFullSalary();
                return fullSalary;
        }
};
```

a) Above class calculates the full salary once you give the basic salary, over-time hours and hourly rate as inputs. Use the following values for each input:

- basic salary =6500
- over-time hours=120
- over-time hourly rate=110

Write a **main()** function which will implement the above class and print the full salary.

(6 marks)

b) Briefly describe what a Default Copy Constructor is.

(2 marks)

d) Write the code segment to implement the Default Copy Constructor in the above **Salary** class.

(4 marks)

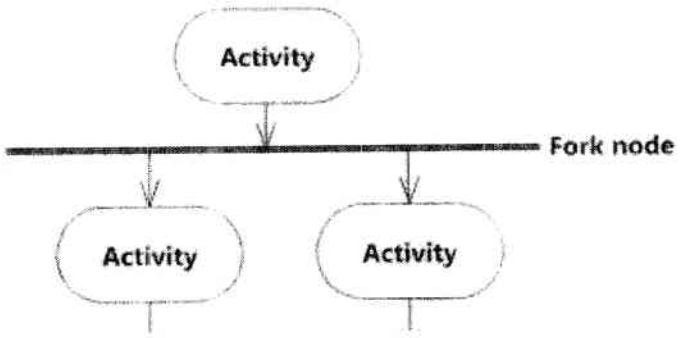
e) Write the code segment for the destructor for the above **Salary** class.

(2 marks)

f) Describe the function of the Scope Resolution Operator.

(4 marks)

g)



Describe concept of the Fork node in UML shown in the figure.

(2 marks)

2. a) Considering the **Salary** class given in above Question 1, describe the **OO** principles which have been used in it.

(5 marks)

b) Draw the class diagram for the **Salary** class given in Question 1 above.

(5 marks)

c) Describe importance of **Abstraction** in your own words.

(4 marks)

d) What is the technical difference between structures and classes in C++?

(4 marks)

e) Briefly describe what an Enumerated type is.

(2 marks)

3. a) Examine the code given below,

```
#include <iostream>
#include <string>

class A
{
  public:
  void say(){
```

```
    std::cout << "A"<<"\n";
  }};

class B: public A
{
  public:
  void say(){
    std::cout <<"B"<<"\n";
  }
};

int main()
{
  A a;
  B b;
  A *c;
  a=b;
  c=&b;

  a.say();
  b.say();
  c->say();
  a.say();
}
```

Write the output of the above program.

(4 marks)

b) Write the output of the above program if the **say()** function in the **B** class is a virtual function.

(4 marks)

c)
```
#include <iostream>
#include <string>
using namespace std;
 class Say
{
  public:
  void say(){
    std::cout << "A"<<"\n";
  }

  void say(string words){
    std::cout << words <<"\n";
  }

  void say(string words, int freq){
```

```
    for(int i=0;i<freq;i++)
    std::cout << words <<"\n";
  } };
```

i. Above **Say** class has three functions named as **say**. Describe the technical concept for having multiple functions with the same name in a class.

(3 marks)

ii. Write the **main()** function which will call **say()** functions in the **Say** class.

(3 marks)

d) Briefly describe the concept of static data in C++.

(3 marks)

e) Briefly describe the concept of **const** objects in C++.

(3 marks)

4 a) Describe the concept of operator overloading with a suitable example.

(5 marks)

b) Name the operators that cannot be overloaded in C++.

(5 marks)

c)
```
#include <iostream>
#include <string>

using namespace std;
class C
{
  private:int a;

  public:
  void seta(int a1){
    a=a1;
  }
};

class D:public C
{
  public:
  void setdata(int d)
  {
     seta(d);
  }
};

int main()
```

```
{
 D d;
 d.setdata(10);
}
```

Consider the classes **C** and **D** above. This program functions without an error. Type **int** variable **a** in class **C** is a private variable. However, a value is set to the variable **a** in class **D**. Explain how this works without an error.

(5 marks)

d) Explain how you would access the overridden function in the base class, using examples.

(5 marks)

5 a) Briefly describe the declaration of an abstract class using code examples.

(4 marks)

b) Name the **three (03)** types of inheritance.

(3 marks)

c)
```
#include <iostream>
#include <string>
using namespace std;

class E
{
  public:void display()
  {
    cout<<"This is E"<<"\n";
  }
};

class F
{
  public:void display()
  {
    cout<<"This is F"<<"\n";
  }
};

class G:public E, public F
{

};
```

Write the code segment to implement the **display()** function in class **G** inherited from class **F**.

(3 marks)

d) Name the default access specifier in C++.

(3 marks)

e) Briefly describe what a friend class in C++ is.

(3 marks)

f) Describe the features of the Singleton pattern.

(4 marks)

---END---