Question 1 (36% of the marks, each of the 6 sub-questions carry equal marks).

i. What is the difference between method overloading and method overriding (i.e. virtual functions) in C++? Give an example to illustrate your answer.

ii. Explain what is wrong with the function overloading code below.

```
int   foo(){return 48;}
double foo(){return 48.0;}
```

iii. What is the output of the program below? Explain your answer.

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

int main()
{
    int a = 1;
    int &b=a;
    b=2;
    cout<<"a="<<a<<endl;
    a=3;
    cout<<"b="<<b<<endl;
    return 0;
}
```

iv. Use an example to explain the concept of a class constructor.

v. What is the output of the program below. Justify your answer.

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

class A
{
public:
    A(){cout<<"Construct A"<<endl;}
};

int main()
{
    A a1;
    A *a2;
```

```
    return 0;
}
```

vi. What is the output of the program below. Justify your answer.

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

class SuperClass
{
public:
    virtual void message()
    {
        cout<<"In Superclass"<<endl;

    }
};

class DerivedClass: public SuperClass
{
public:
    void message()
    {
        cout<<"In DerivedClass"<<endl;

    }
};

int main()
{
    SuperClass *sc1;
    SuperClass sc2;
    DerivedClass dc;
    sc1=&dc;
    sc1->message();
    sc1=&sc2;
    sc1->message();
    return 0;
}
```

Question 2 (C++ Class Construction, 34% of the Marks, the two sub-parts carry 10% and 24% of the marks respectively).

Consider the one dimensional Stochastic Differential Equation below,

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dB_t, \;\; X(0) = x.$$

Where B_t is the standard Brownian motion and $\mu : \mathbb{R}^2 \mapsto \mathbb{R}$, $\sigma : \mathbb{R}^2 \mapsto \mathbb{R}$.

i. Write pseudo-code for the Euler-Maruyama scheme for the numerical solution of the SDE above.

ii. Describe (using code-snippets if necessary) how to implement the Euler-Maruyama scheme using the following three techniques: (a) function pointers, (b) virtual functions and (c) function templates. Explain the advantages and disadvantages of each of the techniques.

Question 3 (Portfolio Optimization, 30% of the marks).

Consider a market that contains only risky assets, and assume that short-selling is allowed. The minimum variance portfolio of risky assets (portfolio A) has mean rate of return $\bar{r}_A = 5\%$ and standard deviation $\sigma_A = 5\%$. Moreover, there is another portfolio on the efficient frontier (portfolio B) with mean rate of return $\bar{r}_B = 30\%$ and standard deviation $\sigma_B = 30\%$. The covariance of the rates of return of A and B is $\sigma_{AB} = -0.01$.

i. What is the portfolio of A and B which has a mean rate of return of 20% (portfolio C)?

ii. What is the variance of the rate of return of portfolio C?

iii. Does portfolio C have minimum variance for its mean rate of return of 20%? Justify your answer.

Answer 1.

1. Method overloading is having functions with the same name but different arguments. Example

   ```
   void area(int a);
   void area(int a,int b);
   ```

   Method overriding is the redefinition of base class function in its derived class with the same signature i.e. return type and parameters. It can only be done in a derived class

   ```
   class A
   {
       public:
           virtual void display(){cout<<"hello";}
   }
   class B:public A
   {
       public:
           void display(){cout<<"bye";}
   }
   ```

2. No. Functions declarations only differ in the return type.

3. a=2, b=3. b is an alias of a. Hence changing one will change the other too.

4. A constructor is a special type of member function whose name is the same as the class name. Constructors are used to initialize the data members of the class. A constructor is called when the object is created. An example with or without arguments should be used to explain the answer.

5. "Construct A". Only one object a1 is constructed, a2 is just a pointer.

6. In DerivedClass. In Superclass. Message is virtual so when call it through a pointer the version of the function used is determined by the pointer type.

Answer 2.
(Sketch) Expect to write three function signatures similar to the ones below:

```
// (I) Function pointer solution: pass the two functions mu and sigma as fu
double solve_Euler_EM(double(*mu)(double t,double x),double(*sigma)(double
   double T, double dt, double X0)
{
    ...

}

class Function
{
   public:
      virtual double mu(double t,double x)=0;
      virtual double sigma(double t,double x)=0;
};
class F1: public Function
{
   public:
        double mu(double t,double x){return ...};
       double sigma(double t,double x){return ...};
};

(II) Virtual Function: pass a class (in this case a pointer)
double solve_Euler_EM(Function *F,double T, double dt, double X0)
{
    ...
}
(II) Function template
template<typename Function>
double solve_Euler_EM(double(Function *F,double T, double dt, double X0)
   {
....
   }
```

Answer 3.

i. Let w be the weight of portfolio A in C. Then, the mean return of portfolio C is given by

$$0.2 \stackrel{!}{=} \bar{r}_C = \bar{r}_A\, w + \bar{r}_B\, (1-w) = 0.05 \cdot w + 0.3 \cdot (1-w)\,,$$

which implies that $w = 0.4$. Thus, C contains 40% of A and 60% of B.

ii. The variance amounts to

$$\begin{aligned} \sigma_C^2 &= \sigma_A^2\, w^2 + 2\sigma_{AB}\, w\,(1-w) + \sigma_B^2\,(1-w)^2 \\ &= (0.05)^2(0.4)^2 - 2 \cdot 0.01 \cdot 0.4 \cdot 0.6 + (0.3)^2(0.6)^2 \\ &= 0.028 \end{aligned}$$

iii. Thus, no portfolio with mean rate of return of 20% has a smaller variance than portfolio C.