Question 1 (33%, each of the 11 subquestions below is worth 3%).

(a) Briefly explain the difference between pass by value and pass by reference when passing arguments in C++ functions.

(b) Give an example of a situation where pass by reference is preferable to pass by value.

(c) Write a function called `interchange()` that interchanges the contents of two variables of type `double`, which are to be passed to the function by reference.

(d) Briefly explain what is meant by multiple inheritance?

(e) What should be the name of a C++ constructor?

(f) How many constructors can be present in a class?

(g) What is the output of this program?

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
        int ran = rand();
        cout << ran << endl;
}
```

Explain your answer.

(h) What is a function template?

- **(i)** Functions that serve as a pattern for creating other similar functions.
- **(ii)** A function inherited from a base class.
- **(iii)** Both (i) & (ii).
- **(iv)** None of the above.

(i) What is a pure virtual function?

(j) What is an abstract class in C++?

(k) What is function overloading?

Question 2 (33%, each of the 3 subquestions below is worth 11%).

(a) Explain (without writing C++ code) how Monte Carlo can be used to compute the delta of a European call option. Does the Monte Carlo simulation need to be run twice in order to compute the delta? Explain your answer.

(b) Let $(S_k(t_1), S_k(t_2), \ldots, S_k(t_m))$ denote the k^{th} sample path generated by a Monte Carlo simulation. You may assume that $t_m = T$. Suppose that the approximate price for a path-dependent option with expiry at time T is given by,

$$\hat{H}_N(0) = e^{-rT} \frac{1}{N} \sum_{k=1}^{N} h(S_k(t_1), S_k(t_2), \ldots, S_k(t_m)).$$

What is the standard error of $\hat{H}_N(0)$. If the numbers of paths are increased from N to $4N$, by how much do you expect the standard error to reduce? Explain your answer.

(c) Explain how the variance of the Monte Carlo method can be reduced using the control variate technique. Give an example of when the control variate technique will be useful in derivative pricing.

Question 3 (34%, each of the 2 subquestions below is worth 10%,18% and 6%).

(a) Derive expressions for the upper and lower boundary conditions of the Black-Scholes equation for a European call option.

(b) Consider the general parabolic partial differential equation below,

$$\begin{aligned}
&\frac{\partial v(t,x)}{\partial t} = a(t,x)\frac{\partial^2 v(t,x)}{\partial x^2} + b(t,x)\frac{\partial v(t,x)}{\partial x} + c(t,x)v(t,x) + d(t,x),\\
&V(T,X) = f(x),\\
&V(t,x_l) = f_l(x),\\
&V(t,x_u) = f_u(x),
\end{aligned}$$

where we look for a solution in $[0,T] \times [x_l, x_u]$ with $x_l < x_u$. Use forward differencing for the time derivative, centered differencing for the first order spatial derivative and a centered second difference for the diffusion term to derive the explicit finite difference method.

(c) Give two disadvantages of the explicit finite difference method.

Answer 1 (33%, each of the 11 subquestions below is worth 3%).

(a) When passing by reference a single copy of the variable in computer memory is shared by the function and the calling program. When the function is called by value a copy of the variable is made in a separate location in memory. The function can see and change the variable but has no access to the original variable.

(b) For example you want to pass a large vector of doubles it might be inefficient to make a copy of the vector. Especially if the call is made many times like in a Monte Carlo simulation.

(c)

```
void interchange(double& x, double& y)
{
        double z;
        z=x;
        x=y;
        y=z;
}
```

(d) Deriving a derived class from more than one base class

(e) Same as class.

(f) There can be multiple constructors of the same class, provided they have different signatures.

(g) It will produce the random number from 0 to `RAND_MAX`. The value of `RAND_MAX`. is platform-dependent.

(h) Answer is (i).

(i) A virtual function with no function body and assigned with a value zero is called as pure virtual function.

(j) A class with at least one pure virtual function is called as abstract class. We cannot instantiate an abstract class.

(k) Defining several functions with the same name with unique list of parameters is called as function overloading.

Answer 2 (33%, each of the 3 subquestions below is worth 11%).

(a) (Sketch). Expected to provide formula for approximating the derivative of the pricing function and explain how the Monte Carlo paths used for pricing can be reused.

(b) (Sketch). Expected to give the formula for the standard error,

$$\sigma_e = \frac{1}{N}\sigma_N$$

Where σ_N is the unbiased estimator for the variance of the price. Then show that $\sigma_{4N} \approx \sigma_{2N}$ and so if the numbers of samples are increased by $4N$ the error will only be halved.

(c) (Sketch). Suppose we want to compute $\mathbb{E}(X)$ and we already know that some other random variable Y is close to X, and that we already know $y = \mathbb{E}(Y)$. The variable Y is called the control variate and the estimate,

$$\mathbb{E}(X) = \mathbb{E}(X - Y) + y,$$

allows us to reduce the error with out increasing the number of sample paths. A good example of this is in the pricing of an arithmetic Asian option, with a geometric Asian option acting as the control variate.

Answer 3 (33%, each of the 3 subquestions below is worth 11%).

(a) (Sketch). Use put/call parity and the fact that as the price approaches 0 a put is worth K, and for very large S a put is worthless.

(b) (Sketch). Let $v_{i,j}$ denote the solution at time i and spatial location j. Then we use

$$\frac{\partial v(t_i, x_j)}{\partial t} \approx \frac{v_{i,j} - v_{i-1,j}}{\Delta t}$$
$$\frac{\partial v(t_i, x_j)}{\partial x} \approx \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta x}$$
$$\frac{\partial^2 v(t_i, x_j)}{\partial x^2} \approx \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{\Delta x^2}$$

Using the above formulas into the BSE we obtain,

$$v_{i-1,j} = A_{i,j}v_{i,j-1} + B_{i,j}v_{i,j} + C_{i,j}v_{i,j+1} + D_{i,j}$$

where,

$$A_{i,j} = \frac{\Delta t}{\Delta x}(b_{i,j/2} - a_{i,j}/\Delta x)$$
$$B_{i,j} = 1 - \Delta t c_{i,j} + 2\Delta t a_{i,j}/\Delta x)^2$$
$$C_{i,j} = -\frac{\Delta t}{\Delta x}(b_{i,j/2} + a_{i,j}/\Delta x)$$
$$D_{i,j} = -\Delta t d_{i,j}$$

(c) It can be unstable for a small stepsize, and it does not scale well to more than 3-4 dimensions.